ESL-TR-92/06-02

LoanSTAR Monitoring and Analysis Program

BUILDING ENERGY MONITORING WORKBOOK

Submitted to the
Texas Governor's Energy Office

by the
Improved Energy Audit Process Task
Jeff S. Haberl, P.I.
Robert López
Robert Sparks

August 1992

ENERGY SYSTEMS
LABORATORY
Department of Mechanical Engineering
Texas Engineering Experiment Station
Texas A&M University System

PREFACE

This building energy monitoring workbook has been prepared for the Texas Governor's Energy Office by the Improved Energy Audit Task of the LoanSTAR Monitoring and Analysis Program. This workbook is intended to be a stand-alone survival guide to acquiring energy use and environmental data in buildings. It includes monitoring procedures and data analysis routines developed for the Texas LoanSTAR program and is copyrighted for distribution in the public domain.

This report has been prepared by Dr. Jeff Haberl, Mr. Rob López and Mr. Robert Sparks. Questions or comments should be directed to Dr. Jeff Haberl, Mechanical Engineering Department, Texas A&M University, College Station, Texas, 77843-3123, (409)845-6065, FAX (409)845-3081, JSH4037@TAMSIGMA, or jhaberl@loanstar.tamu.edu.

DISCLAIMER

TABLE OF CONTENTS.

"Field Data Acquisition Hardware", G. J. Schuster, ORNL Report No. Conf - 8510218, 1985.

"ARCHIVE: Software for Management of Field Data", D.Feuermann, W. Kempton, Center for Energy and Environmental Studies Report No. 216, Princeton University, June 1987.

1-0 INTRODUCTION.

This introduction section describes some of the basic concepts behind the planning and implementation of a building monitoring program. The design of a basic experiment introduced on-off, before-after, simulated occupancy, and test-reference experiments and discusses the types of data measurements that can be used to accomplish the task, including: whole-building and sub-metered measurements, intrusive and non-intrusive data acquisition devices. This section also covers the identification of experimental parameters, and the extent of monitoring. Finally, the basics of before-after monitoring are introduced, including the four levels of monitoring experiments used in the Texas LoanSTAR program. In each of these sections in this introduction references are cited where the reader can go to learn more about the topics that were introduced.

1-1 TYPE OF PROGRAM.

Gathering information and measuring data about energy use and conditions in buildings depends to a large extent on the purpose and objective for performing the work (MacDonald and Wasserman 1989). The following are the categories that represent most building monitoring programs.

Analyzing the Human Factor

One important consideration that has eluded most energy studies is the human factor. A recent study by Komor et al. (1989) showed that consumers, contractors and even engineers may not be receiving enough useful energy consumption information to make the most basic determination (e.g. is a new system performing as expected?). In many instances, consumers do not understand electric demand, and therefore do not have the basis to understand whether a proposed retrofit will save them money. In one instance, even the engineer who designs such systems did not understand how to calculate the electric demand savings.

Building occupants also govern the quality of the indoor environment. Although much has been accomplished in laboratory comfort measurements, field studies seldom measure anything but the most elementary comfort parameters. Energy studies may measure indoor air temperature, sometimes humidity, and only in rare instances ventilation rates or indoor pollutant levels. With improved low-cost measurement and analysis techniques, building scientists are beginning to unravel the complex interactions that link energy use, comfort, and the indoor environmental quality issues.

Environmental Quality Issues

With increasing awareness of environmental quality (both inside and outside), the monitoring of indoor environments has become an issue. Whether it relates to the monitoring of contaminants, lighting conditions, thermal conditions, or other health and safety issues, it is becoming as important to measure why energy is consumed as how much is consumed.

Because many measurement programs make before and after measurements of energy consumption, it is equally important to measure the before and after states of the interior environmental conditions because the two are often directly related.

For example, the installation of lighting reflectors into existing fixtures has caused considerable discussion. Vendors claim that such devices deliver nearly the same illumination to the work plane as the existing non-reflective fixtures with a considerable savings in the energy consumption. Such systems are sometimes installed at no initial cost to the owner. The vendor receives an agreed-upon fixed dollar payment based on the calculated lighting-cost savings for several years. Such a program may or may not be in the best interests of the consumer.

A better way to compensate the vendor would be to measure the actual cost savings which would include the lighting savings, a credit for the reduced cooling load, a credit for the reduced maintenance (there are fewer lamps to re-lamp), a penalty for the increased heating load, a penalty for possible increases in cleaning requirements for the reflectors, and a careful photometric analysis to determine if the proper lighting levels are being maintained (especially over-time when the surface of the reflector becomes dirty). A well designed monitoring experiment could easily accomplish this.

<u>Diagnosing Operational and Maintenance Problems</u>

Inefficient operation of HVAC systems wastes valuable resources, can be difficult to track and rarely receives the attention necessary from understaffed building operators (Haberl et al. 1989). Building operators have a need for focused, timely energy information that is not being met in many facilities. Building operators rarely see the energy impact of their day-to-day actions, and even if they did, without incentives, it is doubtful that they would go the extra mile to save energy. Monitored trend data can provide some relief to building operators when it is presented in a timely fashion in the proper graphic form. Issues about which variables to display, what intervals and which graph types are most effective will still need to be resolved. Another hidden issue is whether or not operators who have been trained with traditional "action-response" methods can be retrained to fully utilize abstract computerized control displays (Zuboff 1988). These critical people-oriented issues and many more like them remain to be answered in the coming years. However, one thing is certain, increasing amounts of computerized trend data are sure to become a reality in the near future for building operators.

<u>Determining System Efficiencies</u>

The evaluation of system or individual components is important to manufacturers, contractors, engineers, utility suppliers, and ultimately consumers. Government agencies can also benefit from improved information about the performance of energy systems and appliances, especially when it is derived from long-term measured data (versus one-time laboratory tests).

Monitoring Energy Savings From Retrofits

Recently, with the advent of affordable, high-powered microprocessors, researchers have begun to unravel the complexities of determining the energy savings from a large number of retrofits by electronically measuring the important parameters of individual retrofits. Actual measured savings from individual building conservation retrofits has, until recently, been too expensive to justify for all but the largest projects. In its place, the energy analyst has had to rely on utility billing data, energy usage estimates, simulations, or rules-of-thumb to calculate the energy savings. Most often, energy conservation retrofits are installed and forgotten with no monitoring or only a cursory look at raw monthly data. Programs which monitor the energy savings from individual retrofits are fast gaining acceptance by utilities, government agencies, and energy analysts. The energy consumer also can benefit from such programs because they will gain (or lose) the most from the purchase of energy efficient appliances.

Evaluating End-use Energy Data

End-use energy data has become an important tool for ascertaining how energy is being used by consumers. With such information (for a statistically significant sample) utilities and regulating agencies can better determine the answers to such complex issues as rebate and bounty programs or third-party conservation financing programs.

Consumption Planning and Load Forecasting

Energy use data are most frequently collected as an aid in planning and forecasting energy production. For the most part such programs are performed by local or regional utility suppliers and serve as a basic tool for assessing system growth, geographic distributions, load characteristics, etc. Local, State, Federal and other government agencies also perform such studies which are used for the purpose of regulating the utility suppliers.

1-2 DESIGNING A MONITORING EXPERIMENT

There are many different ways of designing an experiment to measure the energy efficiency of an HVAC component or to compare one building's energy use to another. Many excellent papers and reports have been written on this topic (Harrje 1982; Frascatero and Lyberg 1983; Lyberg 1987; MacDonald 1989; ASHRAE 1990; Jilar 1990). Most experimental designs can be categorized according to whether they are on-off experiments, before-after experiments, simulated occupancy experiments, or test-reference experiments. The sections that follow are intended to be a brief introduction to the different methods for designing a monitoring experiment and discuss different types of programs, what parameters can be measured and to what extent monitoring can be carried out. This is followed by a basic review of measurement techniques, sensor calibration, and how to analyze methodological errors. In most sections additional material is referenced where the reader can go to learn more.

On-Off Experiments

On-off experiments measure the efficiency improvement of a component or system by simply turning a device on-off and measuring the impact upon the variable of interest. Such experiments, in effect, use the building or system as its own reference. On-off experiments require that the component or system can easily be turned on-off and are therefore limited to systems that can meet this criteria. Extended on-off experiments can be performed by attaching a simple runtime meter to a device, measuring the duration of on-time versus off-time over a given period and relating it to a one-time measurement of power or energy required to turn the device on.

When on-off measurements are being considered for systems or groups of systems in a whole-building configuration the measurement period should either be greater than the time constant for the building or the experiment should be structured to allow measurement of interactions. For example, when considering a lighting retrofit that involves a considerable reduction in the electrical energy consumption it may be necessary to consider the interaction of the lighting system with the heating and cooling requirements of the HVAC system that serves the area. In this case an on-off experiment using sub-hourly measurements (taken on weekends) can provide valuable insight into such interactions and may give an estimate of additional savings attributable reduced cooling loads (or increased heating loads). This idea has been around for some time. It can be shown that a relationship can be deduced by a simple curve fit of an exponential decay relationship to a step change in conditions, an idea that has been confirmed both experimentally and by computer simulations.

Before-after Experiments.

Before-after experiments (which are really one-time only on-off experiments), are usually used when trying to monitor the effects of a permanent change to a building. Before-after experiments are also used when on-off cycling is impossible, for example, when measuring the impact of additional roof insulation. Because before-after experiments are usually a one-time event, careful surveys and planning must take into account system requirements and previous energy consumption characteristics. Appropriate influencing parameters must be identified and measured because before-after experiments usually require normalization for environmental, system and occupancy effects.

Simulated Occupancy Experiments.

Simulated occupancy experiments measure the occupant effect on building energy consumption by varying specific control parameters during an experiment. Simulated occupancy experiments can use on-off or before-after measurement techniques and usually require some type of physiological indices (e.g., comfort, illumination). One example of a simulated occupancy experiment is the use of co-heating units which follow a control pattern that is representative of anticipated occupant behavior.

Test-reference Experiments.

Test-reference experiments involve the comparison of a building to either a similar building, a normalized database or to a simulated building. Test-reference experiments can be applied to most retrofits, including:

To a Similar Building. Test-reference experiments using a similar building require access to two buildings, usually in the same or a similar climate. Construction characteristics, interior environmental conditions and operational parameters must be carefully considered. Such experiments are usually performed when one building has a retrofit applied and one does not. Measurements must identify similarities and differences in the buildings both before and after the experiment.

To a Normalized Database. Test-reference experiments that utilize a normalized database are another way to compare energy usage between a building and what is considered to be "normal" usage. Such experiments seek to measure the impact of a change by comparing the results to those published in a normalized database. Typically, buildings used for comparison must have similar functions (or system requirements), energy consumption (i.e., fuel type), and influencing parameters. Sometimes on-off or before-after experiments are also added to account for varying conditions during the period of the test.

To a Calibrated Model. Test-reference experiments using a calibrated model (e.g., DOE-2, BLAST, ASEAM) are essentially on-off experiments that utilize a simulation model to turn simulated systems (or retrofits) on-off and then measure the effect as simulated by the model. Test-reference experiments usually use forward models, and in certain cases, inverse models or component-based models. Forward models are building simulation programs that model energy use using engineering principals (and varying solutions schemes) and are based on a given set of descriptive building characteristics (i.e., wall U-value, wall area and orientation, HVAC system type, etc.). Forward models typically have a set number of system types and allow the user to input different values for the control variables. Inverse modeling, a type of system parameter identification, simulates a building's energy consumption (or other variable of interest) by identifying a simplified (or aggregate) model from actual performance records (Sonderegger 1978; Rabl 1988; Subbarao 1988, Reddy 1989). Component-based forward models simulate the transient behavior of individual components and allow the user to assemble a system that is representative of portions or all of the system under consideration (Clark 1985; Klein et al. 1976; Sowell et al. 1986). Component-based forward models are the most powerful simulation tool available but, in their current form, are more difficult to master than general purpose models and can require enormous computational resources. This may change however, if current research by Haves and Trewhella (1988) is any indication of things to come.

Each modeling technique has its strengths and weaknesses. Forward models (i.e., DOE-2, BLAST) are most commonly used in test-reference experiments but can present insurmountable problems during the calibration effort (Hsieh et al. 1989) and are limited in the types of systems that can be simulated. Some progress has been made in assembling graphical toolkits to aid in calibrating forward models (Bronson et al. 1992; Haberl et al. 1992). In general these methods use an iterative trial and error method to force the output from a simulation program to match measured data from the building being simulated.

Inverse models are simpler to use, but require measured energy consumption data and measurements of the primary influencing parameters. In some instances inverse models require formulating an appropriate model, deriving a basic set of equations, solving the equations and codifying the solution into an algorithm for each site.

Whole-building or Sub-metering Measurements.

The measurement of building energy consumption primarily involves two categories: whole-building measurements and sub-metering or component-based measurements. Each has its advantages and disadvantages.

Whole-building Measurements. Whole-building measurements are usually accomplished by taking periodic readings at the building boundary. Such measurements can be taken manually on a monthly or even daily basis and the values transcribed into a computer program for analysis. Alternatively, utility meters (gas and electric) can be modified to generate a pulse or analog signal that can then be recorded by a microcomputer at a pre-specified interval (usually hourly) and analyzed with a variety of methods.

Intrusive Sub-metering Measurements. Whole-building measurements can provide a wealth of information about a building's energy consumption. They can show relationships to parameters such as temperature, scheduling, etc., that can lead to a useful predictive model. However, information is often required on specific components or systems which requires sub-metering. Traditional sub-metered measurements have been performed by expensive "hardwiring" of sensors to a dedicated computer where the information can be captured and recorded for later analysis.

Recent developments have introduced new methods for connecting sensors to their attendant computer, including: power line carrier, twisted pair wiring, coaxial cable, infrared, low power radio (FM) and fiber optics. The Electronic Industries Association is developing a Consumer Electronics Bus (CEBUS) which is an open protocol, hierarchical, multi-media, "plug and play" communications media that will greatly reduce the amount of hardwiring necessary o obtain sub-metered information (EIA 1989). The Smart House concept, involving combined communication and power cabling, may also provide an improved technique for gathering information about individual component consumption (Gilmore 1988). The potential for direct FM communications between field monitoring equipment and a centralized database have also recently been announced as has cable-TV based monitoring. Such systems would substitute for telephone modem communications and might prove advantageous where connection by phone is prohibitively expensive. Cellular-phone computer communications have also been proposed for certain applications where such services are available and could prove to be a cost-effective alternative to the traditional "hard-wired" phone line.

Non-intrusive Sub-metering Measurements. Some recent developments have shown that certain sub-metered information can be coaxed from a whole-building signal using high resolution measurements and a continuous statistical analysis, referred to as non-intrusive sub-metering (Hart 1985). Basically, this type of metering consists of a dedicated microprocessor that attaches to the main electrical service to a building and continuously scans the power consumption at a very high sampling rate. The on-off time of individual devices can then be determined by analyzing their high-resolution signature. When combined with individual component consumption characteristics, such measurements can provide sub-metered information about buildings without having to install expensive hardwiring. Currently, this technique has been shown to be effective for residential buildings and is being extended to small commercial buildings (Norford et al. 1992).

1.3 IDENTIFYING THE EXPERIMENTAL PARAMETERS

The design of any monitoring experiment should measure and evaluate both the ingredients and products of the process under consideration. Figures 1-1 and 1-2 are matrices that show typical interactions between the type of program and extent of monitoring versus the design of an experiment. Figure 1-1 shows a matrix of the type of program versus the design of the experiment. Figure 1-2 shows the extent of monitoring versus the design of the experiment.

Measuring the Energy Consumption

Measuring the energy consumption should include: 1) a measurement of the electricity use and demand profiles; 2) measurements of natural gas, fuel oil and other fossil fuel consumption; 3) measurements of thermal energy consumption; and 4) measurements of energy consumed from renewable sources (i.e., solar, wind, geothermal, economizers).

Measuring the Influencing Parameters

The influencing parameters represent significant parameters that influence energy consumption. These parameters can be grouped as: 1) environmental parameters; 2) system parameters; and 3) operational parameters. Wind, ambient temperature, sunlight, and cold water temperature are examples of environmental parameters that can easily be recorded at the site. Operational parameters are occupant related and vary hourly or daily. Typical parameters include occupancy, operating hours, and custodian schedules. System parameters are characteristics that define the installed equipment (e.g., AHU damper settings, thermostat set-points, etc.). System parameters and operating parameters are similar except system parameters are directly related to each subsystem and operational parameters are only indirectly related. Operational parameters tend to change hourly, while system parameters change less frequently, if at all.

1.4 EXTENT OF MONITORING

The design of building energy monitoring experiment must determine the types of data, the variables to be measured, and the time intervals at which variables are recorded in general. The primary types of data available include: database information, point-in-time information, and time-sequenced measurements.

Database Information

Database information is primarily information that is archived in various forms and kept by different types of organizations and agencies. Database information can consist of printed material, charts, graphs, engineering drawings, and specifications. Traditionally it has not been readily accessible in a common computerized format. However, recent trends in the computer hardware industry have dramatically reduced the cost of electronic storage media may hasten the era of the integrated database for building energy information. Database information consists of data from several sources, including: engineering data, as-built, specifications, utility billing data, customer survey records, and other heterogeneous forms of information.

Engineering Data, As-Built, etc. Database information in the form of engineering data, as-built, and specifications are often needed to confirm to the types of systems which exist in a building, wiring runs, control schemes, etc. Such information is commonly used in experiments that are concerned with evaluating systems, analyzing environmental quality, analyzing human interactions and assessing operation and maintenance problems. Information obtained from such documents is usually very helpful to the design of a monitoring experiment and can save many hours of effort tracing wires, and inspecting electrical panels.

Utility Billing Data. Database information in the form of utility billing data typically contains monthly consumption and electric demand data, and sometimes an identifier for the geographical location of the site and SIC code. When combined with daily weather data from a nearby weather station and analyzed with PRISM, the Princeton Scorekeeping Method (Fels 1986), billing data can be used to separate environment-related energy use into several basic categories as shown in Figure 1-3. This figure shows some of the basic models for determining heating, cooling, and base-level energy use from monthly, daily or hourly energy use. Constant, linear, and piecewise linear models can be used to determine heating plus base level, or cooling plus base level. General indicators, or figures of merit, can also be developed for a set of buildings by analyzing Electric Load Factors (ELF), electric demand profiles, and Occupancy Load Factors (OLF) (Haberl and Komor, 1989).

Utility billing data is relatively inexpensive to obtain, is usually easily obtained for a given building, and can provide a general sense of how a building is performing. When combined with a simple phone survey such information can yield a comparative index (i.e., W/ft^2) that can be useful comparing a buildings performance with similar buildings in similar climates.

Heterogeneous Databases. Useful data for certain types of monitoring experiments can often be found in widely dispersed databases (Olsen et al. 1988). Such heterogeneous databases often take the form of publicly available data such as municipal records, state planning data, and local, state and national surveys. Sometimes such data can be found in privately maintained (often unpublished) databases in the form of aggregate utility billing data, utility surveys, and aggregate energy audit results. Most often, such data include monthly consumption and demand information, building description information, and usually are indexed by geographical location, and to a lesser extent, SIC code. For the most part, such data can be obtained for a modest retrieval cost, once the energy analyst knows where to obtain the records of interest.

Point-in-time Information Gathering.

Point-in-time information generally refers to "snapshot" information gathered about a building that can be used to describe "before" and "after" conditions (MacDonald et al. 1989). Usually, point-in-time information is used to gather information about the building description, building occupants, building schedule, system characteristics and control modes, and information concerning recent changes to the building or energy consuming sub-systems. Point-in-time information can be obtained from preliminary surveys or detailed surveys and may include one-time measurements, customer surveys, interviews, and sometimes follow-up surveys.

Preliminary Surveys. Preliminary surveys are typically quite brief, often using phone calls or mail-in responses. Information from preliminary surveys can include details about operating schedules, installed equipment, square feet of conditioned space, and occupant concerns. At best, preliminary surveys can provide rough estimates of such information since many building occupants are usually unaware of the type of heating or air conditioning system in their building, and information obtained may be incorrect (Komor, et al. 1989).

Detailed Surveys. Detailed surveys take the form of the traditional energy audit utilizing information (when available) from previous studies. Typically, such surveys include an instrumented walk-through where information is gathered about the building, its occupants, the energy consuming sub-systems, building zone information, schedule, and occupancy information.

One-Time Measurements During the Survey. Such detailed surveys should, at least, utilize a minimum of instrumentation, including, a camera, a light meter, a thermometer, and a relative humidity meter. In this fashion, existing or "before" conditions can be documented and measured to allow for a comparison to "after" conditions to determine if there has been in improvement (or degradation) of the interior environmental conditions.

Often, in the case of low-cost (or moderate cost) energy conservation retrofits, one-time energy measurements can be taken before and after installation of the retrofit and combined with simple runtime meter measurements to determine the total energy savings. Manual runtime meter readings can then be multiplied by the one-time before-after measurements to determine the energy savings.

Customer Surveys and Interviews. Often, customer awareness, attitudes about energy consumption, or knowledge of energy information can play an important role in discovering opportunities for energy conservation. Such information can be obtained by a simple mail-in or phone survey. However, an interview is often necessary to determine a detailed accounting of such information. Survey techniques vary widely depending on the cost, medium for distribution, intended data entry and analysis method and audience. Interview techniques can also vary considerably. One technique which has gained favor among energy researchers is the ethnographic interview. Ethnographic interviews differ from a strictly "guided" interview in that the informant is allowed to partially guide the discussion and elaborate on the points they discuss (DeCicco and Kempton 1987; Agar 1980).

Follow-up Surveys and Interviews. Obviously, an important aspect of surveys and interviews involves the follow-up survey (or interview). Often, such information can provide completely new insight as to why (or why not) an energy conservation retrofit has succeeded (or failed). Usually, it is best if the initial and follow-up survey can ask the same (or similar) questions in order to ascertain whether or not conditions or perceptions have changed.

<u>Time-sequenced Measurements.</u>

Time-sequenced measurements, or time-dependent data, represent data that changes often enough to warrant a time-series recording. Such measurements can be derived from existing utility billing data, and measured data. Measured time-series data can be accumulated at monthly, daily, hourly and sub-hourly intervals. Such records can then be merged with similar recordings of the influencing parameters such as weather data, operating schedules, etc. at the time of analysis. New research has shown that high resolution data and special purpose data also provides useful information about how a building is consuming energy or whether or not the required environmental conditions are being maintained (Norford, 1992).

Monthly Data. The most common time-sequenced measurement consists of monthly consumption data. Often, such data are readily available in the form of utility billing records. In certain instances, special meters must be installed, read (either manually or electronically) and the information transcribed and prepared for analysis. Monthly data usually must be accompanied by the dates for the period of measurement to allow for weather normalization and/or adjustments for differences in the length of the period.

Daily Data. Daily time-sequenced measurements have gained some acceptance as a useful means of tracking and diagnosing operation problems (Haberl and Vajda 1988; Haberl and Claridge 1987). Such measurements often utilize whole-building utility billing meters, and can be summaries of electronically recorded hourly data or can be read manually and transcribed into a spreadsheet template for analysis. In large institutional complexes, daily measurements have also proven to be helpful in improving whole-campus efficiencies by providing operators with immediate feedback regarding boiler efficiency, make-up water percentages, and predictions of anticipated loads -- information that previously took hours of hand calculations and yet was provided at a fraction of the cost of a fully computerized system (Haberl et al. 1989, Haberl 1992).

Hourly and Sub-Hourly Data. As more powerful, inexpensive microprocessors have become available, the cost of microprocessor-based data acquisition equipment, polling computers and analysis software has also dropped. As a consequence, time-sequenced measurements which consist of hourly and sub-hourly data (1, 5, 10, or 15-minute intervals) are becoming more available to answer detailed questions about all aspects of building operation and control. Often, hourly weather data can be obtained from a nearby station and merged with the site-recorded data. In most locations an analyst can instruct local utilities to install a pulse generating utility meter to provide a signal for a load-monitoring recorder (for a nominal fee).

The size of the data set can become a problem with hourly or shorter measurement intervals when the experiment duration is several months or longer. For a one year period data files approaching one megabyte in size are common. A good set of data preparation tools, or toolkits are very valuable. The public-domain ARCHIVE program developed by Feuermann and Kempton (1987), and the attendant toolkits are examples of excellent software tools for manipulating data in columnar format. Additional public domain data processing software have been produced as part of the LoanSTAR program (a selection of these routines is presented in section 4 of this report). With the use of such tools, data-reduction, data-merging, data-filtering, error-checking, and graph preparation can be reduced to a single key-stroke on most commonly available microcomputers -- a tremendous time saver.

High Resolution Data. The traditional hardwiring of a building to obtain end-use energy data can be expensive and disruptive to the building's occupants; even when advanced FM transmitters or power line carrier systems are used. Recently, an alternative approach has been developed at MIT that eliminates the need for hardwiring (Hart 1985). The basic concept extracts end-use data from high resolution measurements of whole building electricity usage by using statistical analysis to identify on-off times which can then be combined with one-time consumption measurements to calculate end-use energy consumption.

Early results have shown this to be an effective alternative for certain types of end-use experiments. Specifically, those where the components being monitored have a constant energy usage pattern. This technology was originally developed for residential applications and is now being extended into commercial buildings. Commercialization is expected within a few years. Because installation costs can easily represent 2/3 or more of the total monitoring costs, this technology could dramatically reduce the cost of obtaining certain types of end-use energy data. A recent paper by Norford highlights the extension of the original method to commercial and diagnostic tasks (Norford and Mabey 1992).

Special Purpose Data. Another approach to obtaining end-use energy data is to determine which device is running by "listening" to audio signals from a mechanical room and then calculating the energy use in a similar fashion to that used by the MIT high resolution approach. A technique developed by Miller (1989) has been shown to be capable of calculating the simultaneous on-off status of several devices using artificial neural networks for classifying the different acoustic signals. This technique also has potential for diagnosing operation and maintenance problems. Although still in its early stages, the concept of using acoustic pattern recognition (or for that matter visual pattern recognition) opens up an entirely new area of study for building energy analysts.

Before-After Analysis of Energy Conservation Retrofit Savings.

In Figure 1-4 a flowchart of a before-after analysis of energy conservation retrofit savings is shown. For each site before-after point-in-time and time-sequenced information, influencing parameters, and system requirement are evaluated to determine if energy savings match those of the audit estimates. In a program where measured savings are the primary objective corrective measures (if needed) and feedback to owners and operators can be provided.

1.5 BASIC TYPES OF MONITORING EXPERIMENTS IN THE TEXAS LOANSTAR PROGRAM.

Four levels of metering systems have been developed for the Texas LoanSTAR building energy monitoring and analysis program. These accommodate most of the necessary data requirements with the funds available for monitoring retrofits. The levels also are compatible with different hardware available on the market. Table 1-1 contains guidelines for the LoanSTAR energy monitoring levels, and Figures 1-5 to 1-8 illustrate the basic metering experiments.

Level 0: Facility/Whole-building(s) Utility Data.

These data range from monthly consumption data, based on utility bills to weekly or daily utility data. Such data can be useful for separating consumption into heating, cooling, and non-weather related consumption (e.g., water heating) when analyzed with an empirical model.

Level 1: Whole-building and Limited Sub-metered Hourly Data.

Level 1 utilizes one to four channel Data Acquisition Systems (DAS), and will capture hourly whole-building thermal and electric measurements. In some cases, limited sub-metering will also be included. Figure 1-5 displays an example of a typical Level 1 monitoring experiment.

Level 2: Moderate Sub-metered Hourly Data.

This level has all the capabilities of the first two levels and also enables more detailed analysis for identifying the savings from specific retrofits and pinpointing building operational problems. Figure 1-6 shows a typical Level 2 experiment plan.

Level 3: Detailed Sub-metered Hourly Data.

These systems typically include at least 20 channels of data or more. Given current costs, these systems are only expected to be cost effective in large buildings or groups of smaller buildings. Figure 1-7 and 1-8 show Level 3 experiment plans.

1.6 SUMMARY

This introduction section has attempted to describe some of the basic concepts behind the planning and implementation of a building monitoring program. The design of a basic experiment introduced on-off, before-after, simulated occupancy, and test-reference experiments and discussed the types of data measurements that can be used to accomplish the task, including: whole-building and sub-metered measurements, intrusive and non-intrusive data acquisition devices.

The experimental design is highly dependent on the type of program being carried-out. Very different information needs to be gathered at different intervals to cover topics on issues as diverse as the analysis of the human factor in a building, issues about environmental quality, the diagnosis of operation and maintenance problems, measurements of system efficiency and savings from energy conservation retrofits. Almost all of these types of programs are necessary for the integrated planning and forecasting of utility loads.

This introduction also covered the identification of experimental parameters, and the extent of monitoring. In general an experimenter must determine what variables to measure and what intervals to measure those variables at. Finally, the basics of before-after monitoring were covered and the four levels of monitoring experiments used in the Texas LoanSTAR program introduced. In each of these sections in this introduction references were cited where the reader can go to learn more about the topics that were introduced.

TABLE 1-1 *Guidelines for the LoanSTAR Metering Experiments.* This table presents the original first year guidelines for the LoanSTAR metering budget allocations. This table is based on a monitoring budget of 2 - 3% of the retrofit costs and anticipated O&M savings of 5 to 15% of current total energy consumption.

Monitoring Level:	Retrofit Costs:	Annual Energy Costs:	Monitoring Budget:
Level 0: Utility Data	\$20k - \$50k	\$10k - \$30k	\$0
Level 1: 1 - 4 channels	\$50k - \$100k	\$30k - \$60k	\$3k
Level 2: 4 - 20 channels	\$100k - \$300k	\$60k - \$200k	\$10k
Level 3: 20+ channels	\$300k+	\$500k+	\$30k+

FIGURE 1-1 *Type of program versus the design of the experiment.* This matrix shows the relationship between the type of monitoring program and the design of the experiment (Haberl et al. 1990).

	Human Factor	Environmental Issues	O & M Problems	System Evaluation	Retrofit Studies	End-Use Energy Studies	Conservation Planning
On-Off		○		●	●		○
Before-After		●		●	●		○
Simulated Occupancy	●	●		○	●		○
Test-Ref. Similar Building	○	○	○	●	●		○
Test-Ref. Normalized Data Base	○	○	○	●	●		○
Test-Ref. Calibrated Model	○	○		●	●		○
Whole Building Energy & Conditions	●	○	●	○	●		●
Hardwired Sub-Metered	●	●	●	●	●	●	●
Non-Intrusive Sub-Metered	○	○	○	○	○	●	●

FIGURE 1-2 *Extent of monitoring versus the design of the experiment.* This matrix illustrates the extent of monitoring used to accomplish a particular experiment design. The asterisk indicates that utility billing data can be used as a means of checking whole-building and sub-metered experiments.

	Special Purpose	High Resolution	Hourly Measurements	Daily Measurements	Monthly Measurements	Follow-up Survey & Meas.	Customer Survey & Interview	One-Time Measurements	Detailed Survey	Preliminary Survey	Heterogeneous Data Base	Utility Billing Data	Engineering Data, As-Builts.
On-Off	○	○				○	○	●	●	●		○	●
Before-After	○	○	●	●	●	●	●	○	●	●	●	●	●
Simulated Occupancy			●	●	●	●	○	○	●	○		○	●
Test-Ref. Similar Building	○	○	●	●	●	●	○	●	●	●		●	●
Test-Ref. Normalized Data Base	○	○	●	●	●	●	○	●	●	●	●	●	●
Test-Ref. Calibrated Model			●	●	●	●	○	●	●	●			●
Whole Building Energy & Conditions	●	●	●	●	●	●	●	●	●	●		*	●
Hordwired Sub-Metered	●	●	●	●	●				●	●		*	●
Non-Intrusive Sub-Metered	●	●	●	●	●				●	●		*	●

FIGURE 1-3 *Five basic heating, cooling and base-level models.* This figure shows several basic models for determining heating, cooling, and base-level energy use from monthly, daily or hourly energy use. In general, constant, linear and piecewise linear models can be used to empirically determine heating or cooling energy use.

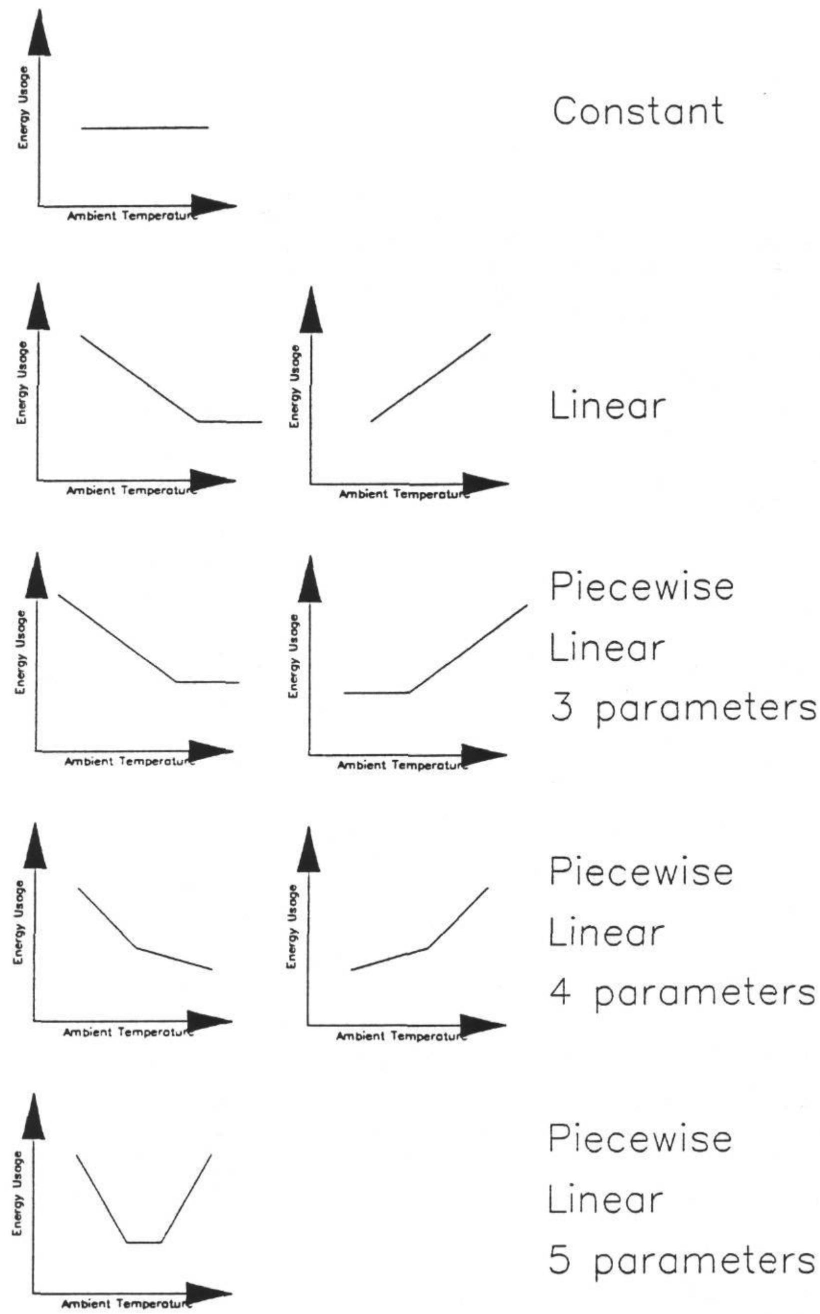


***FIGURE 1-4** Before-after analysis of energy conservation retrofit savings. This* flowchart illustrates the before-after analysis of retrofit savings. For each site before-after point-in-time and time-sequenced information, influencing parameters, and system requirement are evaluated to determine if energy savings match those of the audit estimates. In a program where measured savings are an objective corrective measures (if needed) and feedback to owners and operators can be provided (Haberl et al. 1990).

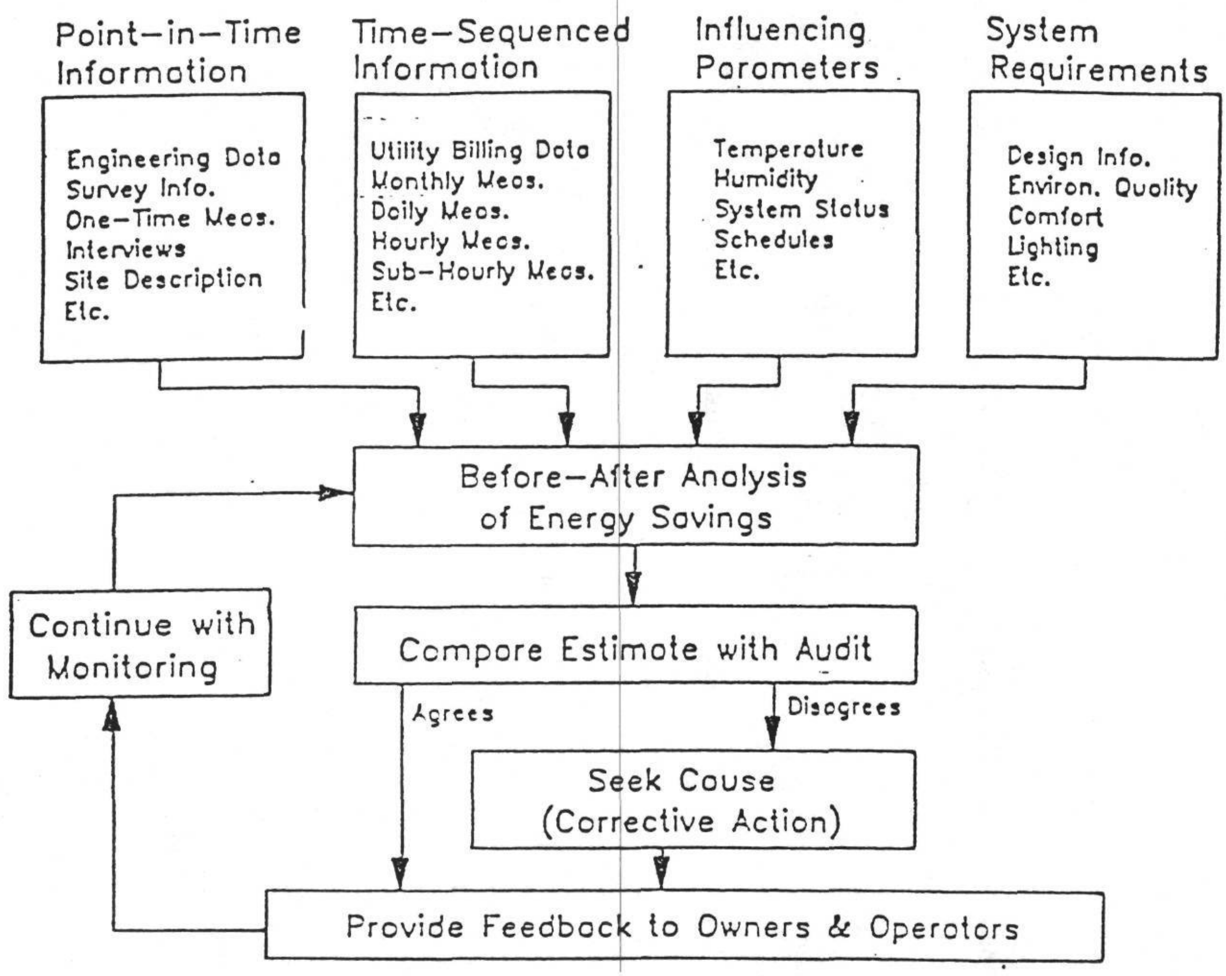


FIGURE 1-5 *Typical LoanSTAR Level 1 Monitoring.* This diagram illustrates a typical Level 1 monitoring setup as installed in the T.S. Painter Building on the University of Texas at Austin campus. This Level 1 monitoring provides whole-building electricity, whole-building chilled water use and whole-building steam condensate use. Channel numbers (e.g. CH 8) are usually included for each site to help identify data channels.

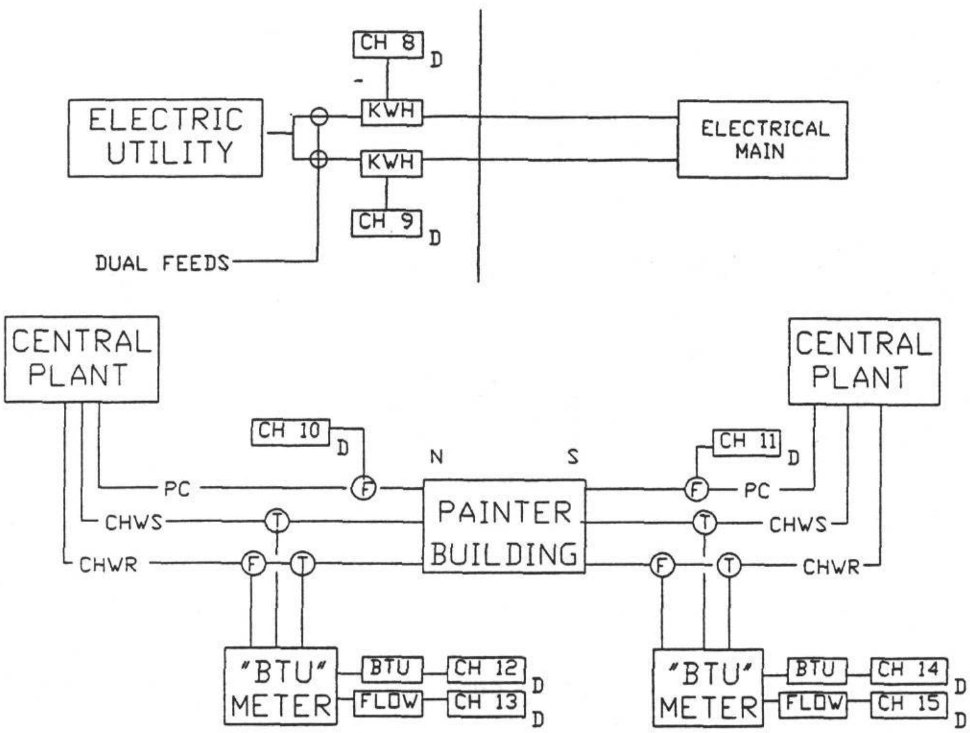


FIGURE 1-6 *Typical LoanSTAR Level 2 Monitoring.* This figure shows a typical Level 2 monitoring experiment plan as installed in the UTC Building on the University of Texas at Austin campus. In this case Level 2 monitoring provides whole-building electricity (kWh), whole-building chilled water use [BTU, flow (F) temperature (T)], whole-building steam condensate use (PC or AC), and sub-metered electricity use for the building's air-handling units (AHUs) and pumps (CHWP).

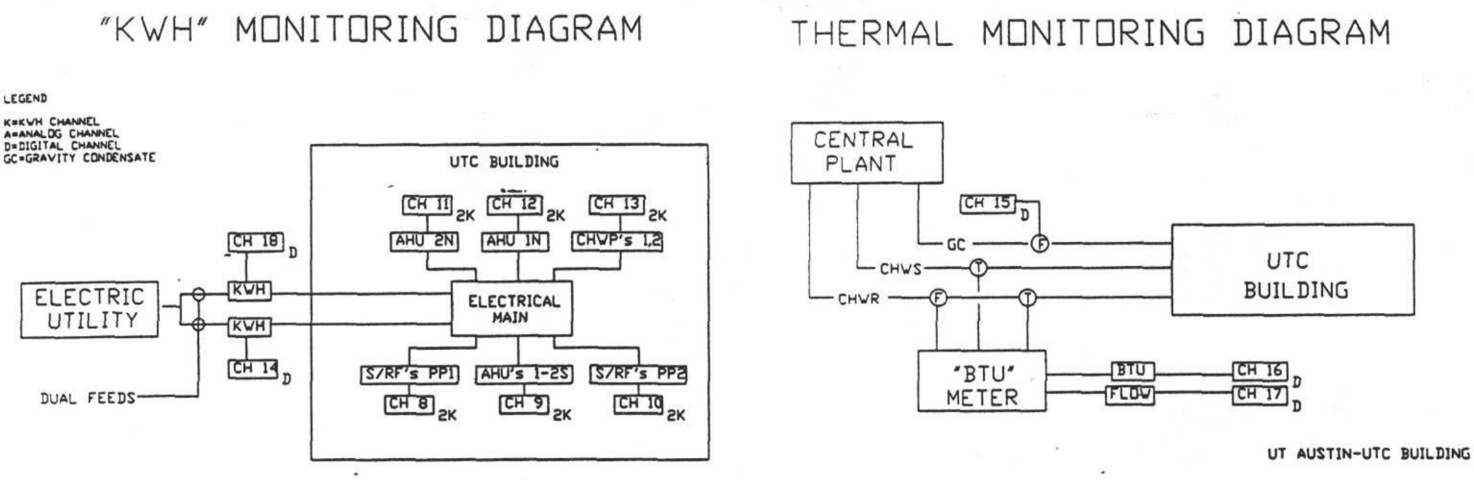
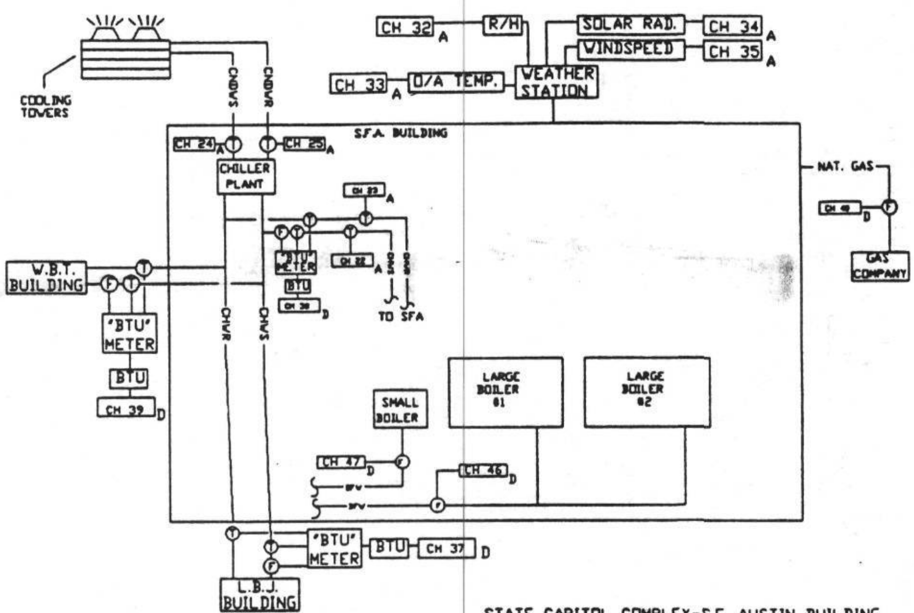
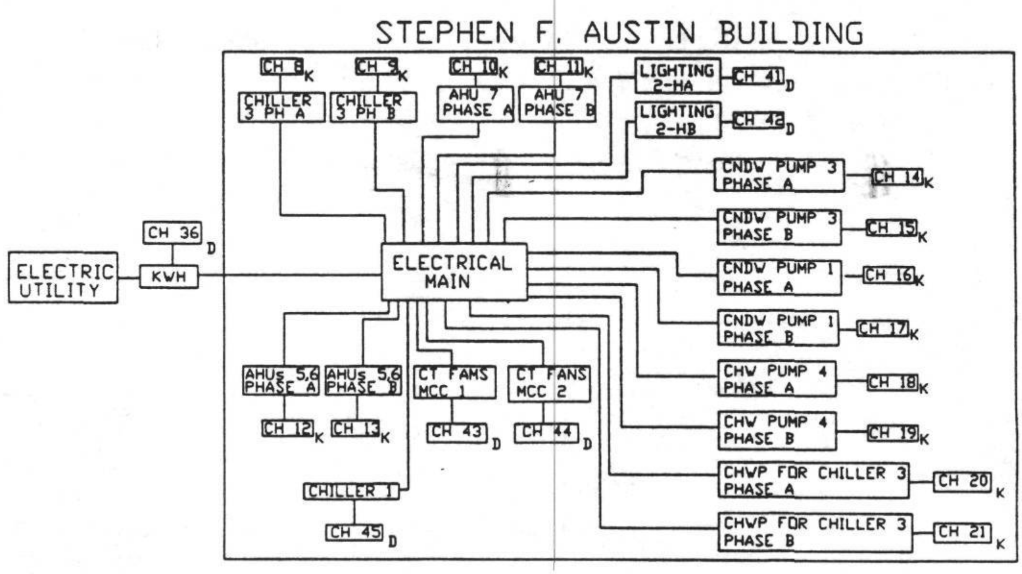


FIGURE 1-7 *Typical LoanSTAR Level 3 Thermal Monitoring.* This diagram illustrates a Level 3 thermal monitoring setup as installed in the S. F. Austin building at the Texas State Capitol complex. In general most Level 3 monitoring experiments are complex and require a metering plan that is specially tailored to each experiment. The S. F. Austin building contains a central boiler/chiller complex that feeds several other buildings and a weather station.



STATE CAPITOL COMPLEX-S.F AUSTIN BUILDING

FIGURE 1-8 *Typical LoanSTAR Level 3 kWh Monitoring.* This diagram illustrates a Level 1 electricity monitoring setup as installed in the S. F. Austin building at the Texas State Capitol complex. The intent of this monitoring plan is to isolate electricity use in the boiler/chiller plant.



STATE CAPITOL COMPLEX-S.F. AUSTIN BUILDING

2-0 REVIEW OF MEASUREMENT TECHNIQUES.

This section provides a review of measurement techniques, including a review of electricity monitoring, temperature measurements, humidity measurements, and flow and Btu measurements. Helpful hints from experiences gained in the LoanSTAR program are also included, specifically, the installation and calibration of sensors, and the analysis of errors.

2-1 BASICS OF ELECTRICITY MONITORING.

The monitoring of electrical energy (a time-varying function) requires a few more steps than the monitoring of electric power (an instantaneous function). In order to measure the energy use of a building or an appliance it is necessary to have a recording meter that measures and records the amount of power used over a period of time. In the early days of electrical metering this was accomplished with some very ingenious mechanical devices. Beginning in the mid-1980s the direct metering of electrical energy through the use of affordable, easy-to-use microprocessor-based field data acquisition systems became a day-to-day reality. However, in order to get a sense of what is involved, it is helpful to know about the basic components--many of which are still in use today.

From the start, the measurement and recording of electrical energy was accomplished with Watt-hour meters, and pulse-initiating telemetering circuits. Then, in the 1970s, this task was simplified somewhat when Watt/Watt-hour transducers became commercially available, and more recently, with the advent of microprocessor technology, digital Watt/Watt-hour meters. This next section reviews the basics of obtaining digital data from power/energy measurements. The reader is referred to the authoritative Handbook for Electricity Metering by the Edison Electric Institute (EEI 1981), and the paper by Schuster (1985) for additional details. A significant amount of material for the next section of this workbook has come from these texts.

The Watt-hour meter.

A Watt-hour/demand meter acts very much like the speedometer and odometer on an automobile. It is composed of a combination of sub-components: a very slow-speed motor whose rotational speed is proportional to the power that passes through it, a magnetic brake to retard the spinning rotor when power is withdrawn from the meter, a series of mechanical registers to record the number of revolutions, and a meter that records the peak electric demand. Basically, as the power is increased, the rotor spins faster and the Watt-hour meter records more revolutions. A measurement of the energy used during any given period of time is then be obtained by subtracting two consecutive Watt-hour meter readings (revolution counts).

Figure 2-1 shows the basic parts of a Watt-hour meter. The mechanical-type Watt-hour meter shown contains a register upon which the energy use is recorded on four decade dials, a electrically conductive rotor (i. e., a spinning motor), retarding magnets, current coils, potential coils, and contact blades to connect it to the metering can. Such meters form the basis for almost all of the electrical metering that is performed in this country. The basic principal upon which they work was set forth in 1884 by Ferraris who showed that torque (rotation) can be produced in an electrically conductive rotor when it is exposed to two alternating-current fluxes in such a way that they produce rotational motion in one direction on the rotor.

In a Watt-hour meter this is accomplished with a lead-lag current and potential coils as shown in Figure 2-2. An electromotive torque (EMF) is induced in a rotating conductor that is proportional to the power passing through the meter. Alternating magnetic flux causes additive eddy currents in a rotating disk that force it to rotate in one direction.

In order to assure that the rotational speed of the disk closely follows both increasing and decreasing power loads, the disk is constantly retarded by permanent magnets which act like brakes that immediately slow the rotor whenever there is a decrease in the EMF. This creates an immediate decrease in the disk rotational speed whenever the power load on the meter (and hence the disk) drops off yet allows the disk to accelerate quickly whenever power is increased. Usually a series of gears connects the disk to decade-type dials on the face of the Watt-hour meter which are used to display the energy used by counting the total number of rotations.

Connecting a Computer to a Watt-hour Meter.

The basic method that is used to generate an electronic time-series data record of the energy that passes through a Watt-hour meter is to send a series of on/off pulses to a digital recorder. Originally, this was accomplished with pen and ink charts which gave way to magnetic tapes that were collected and transferred to a mainframe computer periodically. More recently, such information is recorded by microprocessor-based data acquisition systems where it is transferred (sometimes automatically) over phone lines to a central facility. In almost all cases, the time-series record consists of a record of pulses where each pulse is equal to some predetermined number of Watt-hour disk revolutions and hence the amount of energy that has passed through the meter during that period of time.

In order to accomplish this the energy-accounting gear train of the Watt-hour meter is fitted with either a two-wire or three-wire pulse initiator (either electronic, optical, or mechanical) as shown in Figure 2-3. A simplified schematic of 2-wire and 3-wire telemetering circuits are shown in Figure 2-4. Most Watt-hour meters that are fitted with pulse initiators use the 3-wire configuration. The difference between 2-wire and 3-wire systems is due to physical problems that arose with the original mechanical pulse receivers. If the pulse initiator was not carefully installed and adjusted, there was a chance that it would chatter and over-register the number of pulses. This egregious characteristic was removed by adding a third wire which serves as a latching mechanism since the pulse generator must not only switch on/off but also switch between circuits between each pulse--thus eliminating the chatter. A 2-wire telemetering connection can always be made to a 3-wire pulse initiator by simply using one of the contacts and adjusting the energy/pulse ratio. However, one should always check to make sure that the 2-wire circuit is not chattering. This problem can be eliminated by installing an approximately-sized capacitor.

Measuring demand with a Watt-hour meter.

In most commercial buildings both the electricity energy use and peak electric demand are usually recorded for billing purposes. In order to accomplish this the Watt-hour meter is equipped with a demand meter which retains a reading of the peak electric power level that passed through the meter until the demand reading is reset at the end of each billing period. This was originally accomplished through the use of a thermal demand element which later gave way to a mechanical gear-driven demand meter.

In thermal demand meters a pair of bi-metallic coils are attached to the demand indicator shaft as shown in Figure 2-5. Movement of the shaft is produced by creating a temperature difference between the coils that is proportional to the power measured. This is accomplished by reversing the currents in one heater when compared to the other as shown. Thermal demand meters are still being used today. However, one should be careful when such meters are exposed to widely varying ambient temperatures because the demand readings can be effected by temperature.

A simplified schematic of an indicating-type mechanical Watt-hour demand meter is shown in Figure 2-6. Such a device is actually a "Watt-hour meter within a Watt-hour meter". It records the energy used during a pre-specified period, pushing the maximum demand pointer to a new value when the demand for the current period exceeds any previous period, and then resets itself to zero at the end of a predetermined demand period; beginning the cycle over again. Some versions also include a cumulative-type demand meter that displays the current peak demand and adds the demand to another register each time it is reset.

The Watt/Watt-hour Transducer.

A replacement for the Watt-hour meter became a reality in the 1970s with the commercial availability of the Watt/Watt-hour transducer. This solid state device produced dramatic improvements in the accuracy and stability of electrical metering and paved the way for microprocessor-based electrical power and energy metering. The Watt/Watt-hour meter provides a direct analog or digital output signal that is proportional to the energy being consumed. A functional block diagram of the basic Watt/Watt-hour transducer is provided in Figure 2-7.

Watts are calculated electronically and output as either an analog DC signal or pulsed output that uses a basic time-division-multiplier principle. Conversion of the energy consumption to analog or pulsed output utilizes two different processes. In each process a carefully controlled triangular wave form is compared to a varying sinusoidal wave form to produce a pulse-width and pulse-amplitude modulation. In other words, the width of each pulse is proportional to the input voltage, and the amplitude of each pulse is proportional to the input current. The output from the modulator is a DC current signal that is proportional to the input wattage. The output from the modulator can also be sent to the pulse initiator section to produce a pulse that is proportional to the input wattage. Figure 2-8 illustrates the signal from the comparator and modulator for power factor = 1.0.

The basic unit that is inside integrated solid state digital Watt/Watt-hour meters that are used in certain data acquisition systems utilizes a similar principal as that of the Watt/Watt-hour transducer. In brief, an input reference voltage from a potential transformer (PT) is supplied that provides a signal that is proportional to each of the phases being monitored. This is combined with input current signals to produce digital output signals that are proportional to the energy used by each circuit being monitored. A simplified block diagram of an integrated solid state Watt/Watt-hour meter is shown in Figure 2-9.

The primary advantage with such a multi-channel, integrated, solid state Watt/Watt-hour meter is that it only requires PTs and CTs to attach it to a building's electrical system which eliminates the need for a separate Watt/Watt-hour transducer for each load being monitored, and since it is directly combined with the microprocessor that records the data, it can be re-configured in software for different loads which makes it ideal for portable applications. A more complete description of the circuitry inside the data logger developed for the United States Department of Energy is provided in the paper by Schuster (1985) that is included in the appendix to this workbook.

2.2 MEASURING TEMPERATURE.

The computerized measurement of temperature and humidity is rapidly becoming an off-the-shelf technology. In an average day it is difficult to avoid coming into contact with microprocessor-based temperature measurements in everything from cooking appliances, to automobiles. All of these devices utilize one of several basic techniques for measuring temperature, and it is worthwhile to become reasonably aware of the basic principles of how these measurements are performed. The measurement of temperature (and more recently humidity) relies on principles that were laid-down in the 1800s and later miniaturized, and computerized. This next section provides a brief introduction to the basic techniques that are used to measure temperatures and humidities with computers. The reader is referred to the reference list for additional reading material. In particular the classic textbooks by Benedict (1984), and Doebelin (1990), the NIST (formerly NBS) papers by Hurley (1985), Wise (1976) and Wise and Soulen (1986), and the other references that are listed. This next section borrows heavily from these texts.

The measurement of temperature by a computer is a rather mature technology. In fact, the computerized measurement of temperature has become so reliable that it is quite often used as and indirect method for measuring other quantities such as flow and humidity. Most commonly used computerized temperature measurements use one of four basic methods for measuring temperatures, including: thermoelectric sensors (thermocouples), resistance temperature detectors (RTDs), semiconductor-type resistance thermometers (thermistors), and junction semiconductor devices which are also called integrated circuit temperature (IC) sensors.

Thermocouples.

In 1821 John Seebeck, an Estonian-born physicist, inadvertently discovered the existence of the thermoelectric properties in bismuth-copper, and bismuth-antimony circuits while performing electromagnetic experiments. He showed that when a closed-circuit junction of two dissimilar materials is exposed to different temperatures a small electromagnetic force (EMF) is generated that can be measured as a continuous electric current. Unfortunately, Seebeck misunderstood his findings, thinking instead that he had shown that magnetism was caused by a difference in temperature. In 1834 Peltier, a Frenchman, also observed a thermal effect at the junction of two dissimilar metals. However, it wasn't until 1857 that Lord Kelvin (William Thomson) correctly proved by thermodynamic analysis that the Seebeck effect and Peltier effect were related, and in fact opposite (Threlkeld 1970). This discovery, now referred to as the Seebeck effect, paved the way for the development of thermocouple thermometry.

In thermocouple thermometry, the magnitude of the voltage is dependent on the type of material and the temperature difference. Interestingly enough, when a current from an external source is applied to a thermoelectric circuit, heat is forced to flow from one junction to another. This is the basis for thermoelectric refrigeration (which is often used in a dewpoint humidity sensor.) Most thermocouple junctions (or thermojunctions) are formed by soldering, welding, or simply pressing the two materials together. The most commonly used thermocouple materials are: platinum-rodium (Type S or R), chromel-alumel (Type K), copper-constantan (Type T), and iron-constantan (Type J). Figure 2-10 illustrates the voltage-temperature relation for some of the more commonly used thermocouples.

Most thermocouple applications require some sort of reference junction so that the accurate temperature of one of the thermocouple junctions can be known. Figure 2-11 shows a typical isothermal (constant temperature) reference junction. Basically, by measuring the junction temperature one can compensate for the difference in ambient temperature the temperature that the reference junction is exposed to (versus that which is being measured) and obtain a more accurate measurement. Quite often, one reference junction can be used for multiple thermocouple measurements. In general thermocouples are used when many extremely accurate temperatures are required.

Resistance Temperature Detectors (RTDs).

Electrical resistance in many materials changes with temperature. In some materials this change is very reproducible and therefore can be used as an accurate measure of the temperature. In general materials that display reproducible temperature-resistance properties fall into two classes: conducting (metals) and semiconducting materials. Metallic materials were the first to be developed and have evolved into a class of temperature sensors referred to as Resistance Temperature Detectors (RTDs). The development of semiconducting temperature sensors followed that of RTDs and has been developed into a separate class of sensors called thermistors.

Figure 2-12 displays the temperature-resistance curves for nickel, copper, platinum and manganin. Any slight non-linearity in an RTD signal is usually corrected by applying an empirical polynomial expression that has been derived for a specific sensor. Sensors that use RTD windings come in various shapes and sizes and can be encapsulated and directly exposed to a fluid stream. Most RTDs utilize one of several different types of bridge circuits as shown in Figure 2-13. Of particular interest to measurements in buildings are 3-wire RTD bridge circuits (Figures 2-13c and 2-13d). 3-wire RTDs were developed to compensate for applications where an RTD required a long wire lead that was exposed to varying ambient conditions. This is because three wires of identical length and material exhibit similar resistance-temperature characteristics, and can be used to cancel the effect of the long leads in an appropriately designed bridge circuit. 2-wire RTDs must be field calibrated to compensate for lead length and should not have lead wires exposed to conditions that vary significantly from those that are being measured.

Thermistors.

Thermistors are semiconductor temperature sensors and usually consist of an oxide of either manganese, nickel, cobalt or one of several other types of materials that is milled, mixed, pressed and sintered. One of the primary differences between thermistors and RTDs is that thermistors have a very large negative resistance change with temperature as shown in Figure 2-14. One of the disadvantages of thermistors is that their temperature-resistance relationship is very non-linear. However, when monitoring with a microprocessor, a corrective algorithm can easily be assembled and attached to the raw data stream to yield a linearized thermistor reading. Such algorithms usually have an equation of the form (Doebelin 1990, p. 648):

$$R = R_0 e^{\beta\left(\frac{1}{T}-\frac{1}{To}\right)}$$

where

R = the resistance at temperature T,
Ro = the resistance at temperature To,
β = characteristic material constant,
T, To = absolute temperatures.

Thermistor manufacturers usually publish the appropriate constants for their sensors that fit this equation or one that is similar. Linearization can also be obtained electronically by using a linearization network.

Integrated Circuit Temperature Sensors.

Certain semiconductor diodes and transistors also exhibit reproducible temperature sensitivities. Such devices are usually ready-made Integrated Circuit (IC) sensors and can come in various shapes and sizes. Figure 2-15 shows one manufacturer's junction semiconductor temperature sensor. As indicated in Figure 2-15 IC temperature sensors have a moderately good absolute error over a broad temperature range.

2.3 MEASURING HUMIDITY.

Accurate, affordable, and reliable humidity measurement has always been a time-consuming task for the HVAC industry. Recently, such measurements have become more important in HVAC applications for purposes of control, comfort, and system diagnosis. Although easily explained, psychrometric properties are often misunderstood by the general public (and sometimes engineers). The amount of moisture in the air can be described by several interchangable parameters including: relative humidity, humidity ratio, dewpoint temperature and wet bulb temperature as shown in Figure 2-16.

In brief, relative humidity is the measure of moisture concentration expressed as a percentage of the moisture at saturated conditions. Humidity ratio (lb-moisture/lb-dry air) is the ratio of the weight of water contained in one pound of dry air . Dewpoint temperature is the temperature at which water in the air begins to condense on a surface. Wetbulb temperature, an historic term, is the temperature one would obtain if taking a temperature measurement with a thermometer that had a perfectly "wetted" surface. A psychrometric chart is a graph that is formed by plotting moist air conditions on an x-y graph that has dry bulb temperature on the x-axis and humidity ratio on the y-axis, as shown in Figure 2-16.

Psychrometric terms are interchangeable to some extent. Dry bulb and any one of the following (wet bulb, specific humidity, relative humidity or dewpoint) can be converted to dry bulb and any of the following (wet bulb, specific humidity, relative humidity, dewpoint, enthalpy, or specific volume) using psychrometric routines that are published by the American Society of Heating Refrigeration and Air-conditioning Engineers (ASHRAE 1977). Such a conversion program has been computerized by Texas A&M and is available as public domain software (AIR 1992).

In general most measurements of humidity do not actually "measure" the humidity but rather measure the effect of moisture using an indirect measurement. Dew point measurement includes chilled mirror dewpoint sensors and dewcell sensors. Relative humidity measurements (indirect) include: the evaporation psychrometer, electrical resistance or conductivity, elongation, capacitance-reactance, infrared, radio-frequency, and acoustic measurements. The sections that follow will discuss chilled mirrors, resistance, and capacitance type humidity measurements. For additional material on the evaporation psychrometer, elongation, infrared, radio-frequency, and acoustic measurements the reader is referred to the following references: Cortina (1988), Harding (1982), Huang (1991), Kulwicki (1991), Lee (1988), Leider, et al. (1990), Morrissey (1990), Ross and White (1990), Threlkeld (1970), and Wiesman, (1989).

The measurement and basic understanding of the amount of moisture in the air became a reality in 1911 when Willis Carrier developed his psychrometric formulas and the psychrometric chart (Howell and Sauer 1985). Historically, the measurement of the amount of moisture in the air was first performed with a psychrometer. A psychrometer consists of two thermometers, one that is exposed to the air stream directly (the dry bulb) and one that is wetted (the wet bulb). The basic principle behind the psychrometer is that the temperature of a "wetted" thermometer will always be less than, or equal to the temperature of a "dry" thermometer because of the cooling effect of the evaporation of the water. This temperature difference is called a "wet bulb depression" and is related to the amount of moisture that is already in the air. Psychrometers are best used in a moving air stream. As the air passes over the wet bulb thermometer it evaporates water from the wetted wick which results in a lowering of the temperature of the wetted thermometer. The reading from the wet bulb and dry bulb thermometers are then used to determine the other psychrometric properties by using the appropriate algorithm.

A sling psychrometer consists of a dry-bulb thermometer and a wet-bulb thermometer that are mounted together on a rotating blade that is attached to a handle with a pivot (Figure 2-17). The air movement is obtained by manually whirling the sling psychrometer. Over time, the use of sling psychrometers for measuring humidity in an HVAC system has gradually given way to other methods of measuring humidity. This is because of the significant amount of maintenance that is required to maintain the wetted wick, and keep it free from contaminates, etc. Sling psychrometers remain popular with HVAC engineers as a means of reliably "field checking" their instrumentation. With proper use sling psychrometers can reliably measure wet bulb temperatures to about $\pm$ 1F given air temperatures of 35 to 200 F and humidities of 20 to 80%. Most of the problems that occur with a sling psychrometer are due to human error, and/or inappropriate applications (e.g. freezing conditions, + 80% RH, exposure solar radiation, etc.). The reader is referred to the text by Threlkeld (1970) for additional details.

<u>Chilled-mirrors humidity measurements.</u>

One of the methods of determining the amount of humidity in the air is to simply expose the air stream to a carefully controlled cooled-surface that is at the dewpoint temperature -- the temperature at which moisture begins to condense from the air stream. Careful measurement of the surface temperature will yield the dewpoint temperature which can then be combined with a corresponding dry bulb temperature to produce the intended psychrometric properties. Such a sensor is called a chilled mirror dewpoint sensor.

Figure 2-18 is a schematic of a basic chilled-mirror dewpoint sensor. In such a device the presence of moisture in an air sample is measured by constantly checking for moisture on a refrigerated surface, and then reporting the temperature of the surface. This is usually accomplished optically as shown. The cooling effect is provided by a precisely-controlled thermoelectric refrigerator (also known as the Peltier-Seebeck-Joulean-Thompson conduction effect) -- basically a thermocouple with a current passed through it which causes one of the bimetallic junctions to heat-up and the other to cool-down. In a chilled-mirror dewpoint sensor the output signal is the surface temperature, or dewpoint temperature. When used in extremely humid conditions, such a device sometimes requires a "heated, chilled mirror" which allows for a continuous dewpoint reading, or near saturation conditions.

Chilled-mirror dewpoint sensors are routinely used whenever an extremely accurate humidity reading is needed. However, such sensors require constant maintenance and must be carefully installed to assure that the mirror surface remains clean, and is maintained at the same pressure as the sampled air stream. The pathway of the air to and from the sensor must also be carefully chosen to avoid problems with adsorption and desorption in the tubing walls.

Resistance-type humidity measurements.

The electrical conductivity of certain hygroscopic materials varies in proportion to the amount of moisture absorbed by the material. In certain materials this occurs in a repeatable fashion and can be used to measure the relative humidity of the surrounding air. One of these sensors, known as a Pope cell-type sensor, utilizes a thin layer of sulfonated polystyrene which has been placed on an insoluble surface. An electrically conductive layer is then bonded to the resin and electrodes are attached to facilitate the measurement of the difference in electrical resistance as shown in Figure 2-19. Such a device exhibits a non-linear change in resistance as moisture is absorbed by the hygroscopic resin, varying from a few megohms to about 1,000 ohms at 100 percent saturation. The non-linearities are usually handled by making the sensor one leg of a Wheatstone bridge network. Care must be taken to keep the surface of a Pope-type sensor free of contaminants (such as an oil film) that could interfere with the absorption and desorption of moisture.

Capacitance-type humidity measurements.

Aluminum-oxide capacitance-type sensor. This sensor consists of an aluminum strip that has a porous oxide layer deposited on its tip underneath a *very thin* gold layer as shown in Figure 2-20. The aluminum base, the oxide layer and the gold electrode act as an aluminum oxide capacitor that has a capacitance that varies with the amount of moisture in the porous oxide layer. As water vapor passes through the very thin gold layer it is absorbed and desorbed by the porous oxide layer and changes the electrical capacitance and resistance of the layer. This sensor is known as a Jason-type hygrometer, and can be used to obtain reasonably good humidity measurements provided that it remains below 85% RH. At conditions above 85% RH such a sensor saturates and irreversibly drifts.

Thin-film capacitance-type sensors. Other capacitance-type sensors have been introduced that are similar in operation to the Jason-type hygrometer; one of these is the thin-film polymer capacitance-type sensor. A thin-film polymer humidity sensor uses a polymer in place of the porous oxide to absorb and desorb moisture. The polymer is usually mounted between a rigid aluminum base and another electrode (usually a thin gold film). The polymer is carefully chosen so that it exhibits a change in capacitance with a change in absorbed moisture. In one type of sensor, this changing capacitance changes the frequency of an oscillating circuit which in turn is changed into a varying voltage or current that is proportional to the moisture present. Figure 2-21 shows a thin-film capacitance sensor.

2.4 MEASURING FLOW, AND BTUS.

In many situations, whole-building Btu measurements are needed for a building or group of buildings. Most often this requires that accurate measurements of liquid flow and temperature, usually at the service entrance to the building. Even in cases where steam flow must be measured in a closed loop, it is easier (and much safer) to measure the returning liquid condensate than to measure the live steam as it enters the building.

Choosing a flow meter for a particular application requires a knowledge of what type of fluid is being measured, how dirty or clean that fluid is, what the lowest expected flow velocities for that fluid are and what type of budget one has available as shown in Table 2-1. This next section discusses the common liquid flow measurement devices that are used in conjunction with temperature measurements to determine the thermal energy in a fluid flow.

Flow measurement, like humidity measurement, has greatly improved with the advent of the microprocessor. However, many of the basic sensors remain the same as those that have been in use for many years. In general, these sensors can be grouped into four different types of meters, including: differential pressure flow meters (e.g., orifice plate meter, venturi meter, pitot tube meter), obstruction flow meters (e.g., variable-area meter, positive displacement meter, turbine meter, tangential paddlewheel meter, target meter, vortex meter), non-interfering meters (e.g., ultrasonic meter, magnetic meter), and mass flow meters (e.g., coriolis mass flow meter, angular momentum mass flow meter). The sections that follow will briefly describe each of these meters. Additional details can be found in Doebelin (1990), Baker and Hurley (1984), and Miller (1989).

<u>Differential pressure flow meters.</u>

Differential pressure flow meters are probably one of the oldest techniques for measuring flow in pipes. The basis for their use is Bernoulli's theorem which converts differences in flow rates into differences in pressures which can be easily measured. There are three basic types of differential pressure flow meters, namely, orifice plates, venturi meters, and pitot tubes. In general these meters are made to standard dimensions and are considered very accurate if installed and maintained properly.

Orifice plate meters. Figure 2-22 illustrates an orifice plate differential-pressure flow meter. In such a constant-area, variable-pressure-drop meter (or obstruction meter) the presence of the flow restriction causes an acceleration in the flow and a corresponding pressure drop which varies predictably with the flow. In general, the volumetric flow rate (Q) for one-dimensional, incompressible, frictionless fluid, w/o heat transfer or elevation change can be determined by converting the differential pressure signal with a variation of the following equation (Doebelin 1990, p. 566):

$$Q = \frac{A_{2f}}{\sqrt{1-(A_{2f}/A_{1f})^2}} \sqrt{\frac{2(p_1 - p_2)}{\rho}}$$

where

$A_{1,f}$, $A_{2,f}$	= cross-sectional area where P_1 and P_2 are measured (Figure 2-22),
ρ	= the fluid mass density,
p_1, p_2,	= the static pressures at locations 1 and 2.

Although orifice plate meters are simple in design they produce a significant amount of pressure loss which must be considered in the system in which they are being installed. The installation of an orifice plate meter (venturi meter or flow nozzle) is often impractical because it requires the complete drain-down of the system in which it is being installed.

Venturi meters. Figure 2-23 displays a cross-section of a venturi flow meter. In general the venturi flow meter (or venturi tube) utilizes the same flow-pressure relationship as that of the orifice plate meter. However, the design and manufacture of the meter body is somewhat more complex. Instead of an orifice plate to produce the restriction,the venturi uses a smooth throat. The main advantage with a venturi meter is that it offers a lower pressure drop while providing the same accuracy as an orifice plate meter. A flow nozzle (not shown) is a slightly more complex meter that actually combines an orifice plate with a venturi tube meter. Nozzle meters tend to be preferred in steam and high pressure gas metering installations.

Pitot tube meters. The pitot tube meter utilizes the same basic design that is used in an aircraft to measure the speed of the craft. In an ideal pitot tube the velocity of the fluid at the entrance to the pitot tube can be determined for a one-dimensional, incompressible, frictionless fluid using (Doeblin 1990, p. 528)

$$V = \sqrt{\frac{2(p_{tot} - p_{stat})}{\rho}}$$

where

V	= the flow velocity,
ρ	= the mass density of the fluid,
P_{tot}	= the total free stream pressure,
P_{stat}	= the static stream pressure.

Figure 2-24 is a basic cut-away diagram of a pitot tube used for a point velocity metering application typical of one used in an HVAC duct. When a velocity profile is needed for an entire cross-section, a single point pitot tube can be transversed across the duct or an averaging array can be installed that combines the signals from multiple pitot tubes into a single signal for the entire duct.

In applications where clean fluid flow is expected an averaging pitot tube like that shown in Figure 2-25 can be applied. This "interpolating" pitot tube provides highly accurate flow measurements by using impact ports that are precisely located for a given pipe size to duplicate the calibration conditions in which the pressure differential constants were determined. Such a meter imposes a low pressure loss and can be inserted into an existing pipe without having to drain the system through the use of a welded wet-tap.

Non-pressure-differential obstruction flow meters.

Several additional types of obstruction flow meters have been developed that are capable of providing a linear output signal over a wide range of flow rates, often times *without* the severe pressure-loss penalty that is incurred with a orifice plate or venturi meters. In general these meters place a much smaller target, weight or spinning wheel in the flow stream that then allows the velocity of the fluid to be determined by the force on the meter body (target or variable area meter), and by the rotational speed of the meter, (turbine, paddlewheel meters).

Variable-area meters. The variable-area meter, also called rotameters, measure flow by vertically lifting a weighted float (or spring loaded float) a proportional amount equal to the velocity of the fluid. One type of variable-area flow meter is shown in Figure 2-26. In general variable-area flow meters utilize a tapered shaft to maintain a constant differential pressure across the float. As the flow increases the tapered float rises up the tapered shaft, thus allowing more cross-sectional area for the flow. Variable-area meters are often used as visual displays for flow rates and can have magnetic-type electronic pickups or an extension tube attached to the float that rises and falls with the flow.

Positive displacement meters. Positive displacement meters are meters in which a fluid is forced to repeatedly fill and empty a chamber. Each filling, and the subsequent recording of the filling, indicates that a specific volume of fluid has passed through the meter. In Figure 2-27 a nutating-type positive displacement meter is shown. Positive displacement meters are usually applied to measurement low velocity of slow applications. They are often used when it comes to dispensing fluid commodities such as gasoline because they are repeatable, accurate, and can be incorporated into a compact enclosure since there is no need for a developed length of flow. The primary element can be a reciprocating piston, nutating disk, lobed impeller, sliding vanes or rotating vanes.

Turbine meters. Turbine meters measure fluid flow by counting the rotations of a rotor that is placed in a flow stream. Turbine meters can be an axial-type or insertion-type. Axial turbine meters (Figure 2-28) usually have an axial rotor and a housing that is sized for an appropriate installation. Insertion turbine meters utilize a welded "wet tap" (usually about 2" diameter) that allows for the axial turbine to be inserted into the fluid stream and use the existing pipe as the meter body. Insertion turbine meters can be installed directly into fluid-carrying pipe without having to drain the pipe.

Tangential paddlewheel meters. Figure 2-29 shows two profiles of a tangential paddlewheel meter. Like the insertion turbine meter (and the averaging pitot tube meter) such meters can be installed with a wet tap directly into a fluid carrying pipe without the need for draining the system. In a tangential paddlewheel meter flow is determined by counting the rotations of a tangential rotor in a flow stream. In both turbine flow meters and tangential paddlewheel meters the actual counting is accomplished by converting rotations from either a magnetized or a non-magnetized rotor into an equivalent frequency or pulse. The exact technique that is used is usually jealously guarded by each manufacturer.

Target meters. An illustration of a target meter is provided in Figure 2-30. In a similar fashion as insertion turbine meters, target meters (also called drag force meters) place an obstacle in the path of the fluid. However, instead of employing a rotor a target meter measures the flow of a fluid by measuring the stress on stem that supports the target. This is usually accomplished with a resistance-type strain-gauge that is placed near to the point where the support stem is attached to the meter body. Higher rates of flow cause the supporting stem to bend backwards which is measured by the strain gauge.

Vortex meters. Vortex meters (Figure 2-31) utilize the same basic principle that makes telephone wires oscillate in the wind between telephone poles. This effect is due to oscillating instabilities in a low field after it splits into two flow streams around a blunt object. The measurement of the fluid flow takes advantage of the principle that the frequency of the oscillations is linearly proportional to velocities above $R_e \geq 10{,}000$. The oscillating frequency usually varies from 1 to 500 Hz and can often be measured by piezoelectric, strain-gauge meter, or by magnetic pickup. Vortex meters can effectively be used to measure either liquids, gases or steam.

Mass Flow Meters.

In certain measurements the mass or mass-momentum of a fluid needs to be measured rather than the volumetric velocity. The earliest mass flow meters combined a volumetric flowmeter with a continuous weighing device. More recently, meters have been developed that measure both quantities, specifically, the coriolis mass flow meter, and angular momentum mass flow meter.

Coriolis mass flow meters. Coriolis mass flow meters are essentially vibrating gyroscopes that use principles of angular momentum to measure the mass of a fluid passing through them. Coriolis meters are obstructionless, and are basically insensitive to viscosity, pressure and temperature differences. Such meters can be used to measure homogeneous liquids and mixtures of liquids and gases. Figure 2-32 illustrates the basic principle behind a Coriolis mass flow meter. The mass-momentum measurement of a fluid mixture is measured as small differences in the oscillating frequency of a vibrating u-tube.

Angular momentum mass flow meters. One of the earliest mass flowmeters was developed to measure fuel in the aircraft industry and utilized the moment-of-momentum principle of turbomachines. Simply stated the torque exerted on a fluid by an impeller wheel is dependent on the mass flow rate through the wheel, and the inlet-outlet tangential velocity difference. Figure 2-33 illustrates one kind of angular-momentum mass flowmeter. In this device an angular momentum is first imposed on a fluid stream by a rotating impeller. The spinning fluid stream is then passed into a similar non-spinning impeller that is held in place by a calibrated spring. The force on the spring is then proportional to the mass flow rate.

Non-interfering flow meters.

In all of the previously mentioned meters some interference with the flow stream was necessary to extract a measurement. Recently, a relatively new class of meters has been developed that are able to extract a measurement without placing an obstruction into the fluid stream.

Ultrasonic flow meters. Ultrasonic flow meters measure clean fluid velocities by detecting small differences in the transit time of sound waves that are shot at an angle across a fluid stream as shown in Figure 2-34. Various designs have been developed that utilize multiple pass, multiple path configurations. Accurate clamp-on ultrasonic flow meters have been developed that now facilitate rapid measurement of fluid velocities in pipes of varying sizes. Recently, an ultrasound meter that uses the Doppler principle in place of transit time has been developed. In such a meter a certain amount of particles and air are necessary in order for the signal to bounce-off and be detected by the receiver.

Magnetic flow meters. Magnetic flow meters measure volumetric fluid velocities by magnetic induction as shown in Figure 2-35. When a moving conductive fluid is exposed to a strong magnetic field a potential difference is induced that is proportional to the average velocity. Because both the magnetic field and induced voltage measurement can be obtained without obstructing the fluid flow, such meters have proven useful for measuring flow in a wide range of applications, including, corrosive fluids, slurries, and non-Newtonian fluids.

Btu meters.

The measurement of thermal energy used in a building's heating or cooling system often requires the measurement and recording of Btus. A Btu meter requires both flow measurement and a temperature difference to continuously solve the following relation

$$Btu = C\int M(T_1 - T_2)dt \quad .$$

where

C = the specific heat of the fluid,
M = the mass flow of the fluid, and
T_1, T_2 = the supply and return temperatures.

Prior to the widespread availability of microprocessors this was accomplished by mechanical means as shown in Figure 2-36. In any Btu meter the energy content in a liquid flow requires the subtraction, multiplication and totalization of variables. In the ingenuous mechanical device in Figure 2-36, the subtraction was accomplished by connecting one temperature measurement to the pointer and one to the scale, which produced a temperature difference. Then, through a series of cams, this temperature difference was used to multiply the flow rate from the flow meter into an accumulator that measured Btus as shown. This same basic principle (subtraction, multiplication and totalization) is used today in microprocessor-based Btu transducers.

2.5 INSTALLING AND CALIBRATING SENSORS AND INSTRUMENTS.

Any analysis based on measured data relies on the accuracy of the measurements and how well they portray the true events that transpired. Measurements taken in buildings are no exception to this condition. In fact, since monitoring programs can extend over several years, calibration and recalibration of the instruments and sensors used in the experiments becomes an important aspect of the experimental design.

Installing sensors.

The installation of measurement sensors is a critical link in any monitoring experiment. Although there is no teacher like experience, here are a few tips that could have saved LoanSTAR installation crews a lot of headaches.

Electrical energy measurements. Make sure that the intended end-use loads are actually on the circuit or panel being monitored. Check and double-check field information from local electricians, etc. against measured loads. In any electrical power measurement the correct alignment of phases is necessary to assure proper metering. Often this can require special detective work to determine if the "A" phase is really the "A" phase on all panels, if even marked at all. Match CTs with the anticipated load. Try not to over-load any CTs or severely under load a CT. Make sure to match the digital output frequency (pulses/second) with the receiving ability of the data acquisition system. Although this is not a problem in most applications, sometimes a Watt-hour transducer can be putting-out 100 pulses/second when the data acquisition system can only "see" 50 pulses/second. Make sure that the polarity of each CT is correct, especially when using summing modules that can hide a reversed CT that is reading negative.

Analog measurements. Avoid ground loops at all costs. Mechanical rooms are notorious for having intensive magnetic and electrical fields from motors and transformers. Make sure to check each installation for ground loop current before connecting to the data logger. Although some data loggers can accept 3-wire RTDs, many cannot. Try to minimize long 2-wire RTD leads whenever possible. Pre-calibrate sensors with lead wires permanently attached before installing in the field when possible and then don't cut the lead wires. Use matched lengths whenever possible on any DT measurements. Always use shielded wire and ground properly only at one point to avoid ground loops.

Digital sensors. Check and double-check any manufacturer's pulse constant. Whenever possible use an independent method to check the value of a pulse constant. For example, a KYZ pulse should always be checked against the monthly utility bill. Make sure that you have checked for overcounting of pulses on any 2-wire telemetering circuits. This can occur whenever contact closure "bounces" between on and off states, which can be seen on an oscilloscope or an analog ohm meter. The simple remedy to this is to install an appropriate capacitor to "debounce" the circuit.

Flow measurements. Check flow velocities in a pipe before specifying the final flow metering equipment. Many manufacturers now use a 2" wet tap to install their flow sensor in a "live" pipe. Most flow devices are capable of measuring flow rates at velocities above 3 feet per second. However, in installations where flow rates are in the 1/2 to 3 feet per second range, pre-calibration tests should be performed on typical flow sensors in the intended pipe size to determine the suitability of a particular meter for a given installation. Insertion rotors should be carefully installed at the manufacturer's recommended depth, and upstream flow obstructions should be avoided at all costs.

Check the logger's power supply. Often, access to mechanical rooms is provided to outside contractors who may or may not be aware of a metering experiment. In some cases power to a logger can be inadvertently disconnected to provide power for another device, or switched-off without knowing it. Although simple in concept, providing reliable power to a logger located deep in a mechanical room can be a frustrating task. Also, before using any data logger, make sure that bench tests have been performed to determine the duration that the logger can go without power. The impact of short power interruptions should also be determined in advance (i.e., data loss or communication loss). Short power interruptions can cause microprocessors (in the modem) to lose its programming which can be a expensive problem for a remote site located hundreds of miles away. This problem is easily remedied with 120 VAC circuitry that always assures that any power interruption is 10 seconds or more which is ample time for the logger and modem to reset and maintain communication. Always isolate all power connections, data lines, and phone lines with fuses and/or spike protectors.

Calibration of the instrumentation.

In contrast to the computer industry (where new and better equipment is introduced almost daily), the methods for calibrating instruments commonly used in building monitoring experiments have changed little over the years. A significant body of literature documents standard methods for calibrating instruments. A list of documents describing such procedures is included in the bibliography.

Any serious monitoring program should use the procedures described by a certified standards institute, for example, those published by the National Institute of Standards and Technology (formerly the National Bureau of Standards) or should at least have access to such a facility where instruments can be periodically sent in for calibration.

In the LoanSTAR program a calibration facility has been established to calibrate sensors used in the program. Table 2-2 and 2-3 provide a listing of the different sensors that need calibration in a large-scale monitoring program. The measurement of building energy use and the accompanying influencing parameters often requires the measurement of liquid flow rate, temperatures, hydraulic pressures, air pressures, humidity, electrical power, electrical current, electrical voltage, air-flow rates, rotational speed, solar radiation and illumination levels.

In the LoanSTAR program the calibration of several sensor types has proven to be important, including flow measurement and humidity measurement. Figure 2-37 provides a photograph of the LoanSTAR dynamic-weight flow meter testing facility. In this facility, the accuracy of flow meters in 3 to 10 inch pipes can be tested against both a calibrated orifice plate and a dynamic-weight measure. This allows for the signal from a flow sensor to be tested over a range of fluid velocities varying from 1/2 to 10 feet per second. Figure 2-38 provides an example of the results from testing magnetic and non-magnetic tangential paddlewheel flow meters in a 6 inch pipe. Although such meters provide reasonable accuracy for flow above 3 feet per second, one should be careful to make sure that the velocities are not falling below about 2 feet per second.

The measurement of relative humidity is also problematic. In the LoanSTAR calibration laboratory humidity calibration is performed with saturated salt solutions as specified by NIST. Figure 2-39 provides an example of typical results that can be obtained by comparing the relative humidity from a manufacturer's sensor against the humidity predicted by a saturated salt solution.

Recalibration of Instrumentation.

Instrumentation should also be recalibrated periodically. Unfortunately, this can add substantially to the cost of a monitoring program. Several helpful rules-of-thumb are: 1) When in doubt...calibrate; 2) Calibrate those sensors that are measuring the most critical variables more often (i.e., large thermal flows, whole-building weather-monitoring sensors, etc.); 3) Use as many redundant sensors and sum-checks as often as your budget will permit; and, 4) Periodically recalibrate your calibration instruments.

2.6 ANALYZING ERRORS.

Like death and taxes, we will always have "errors" with us. However, with a concerted effort, the most damaging (and potentially embarrassing) ones can be avoided.

Blunders and Mistakes

Even the best experimenters make mistakes and occasionally a major blunder. Such occurrences are usually apparent from results that are not even reasonably close to the expected value. A simple range check will often suffice to catch most blunder-type errors. Occasionally, slightly more complicated statistical measures (i.e., mean, standard deviation, or goodness of fit) need to be employed.

Systematic Errors

Systematic (often called "bias" errors) are not always detected by simple range checks or other statistical measures. Often systematic errors result from instruments or sensors that have drifted out of calibration. One method for trapping systematic errors is to compare current measurements to estimated values. Corrections to the data that compensate for systematic errors can be made once the type and extent of error are known. The experimental accuracy is therefore dependent on how well systematic errors are controlled.

Random Errors

Random errors represent differences in the measured and true values that cannot easily be removed. They are generally due to sensor or other instrument limitations. Such fluctuations should be minimized within the available budget. The size of the remaining errors must be found to determine the experiment precision. Random errors can be reduced by carefully planning the experiment and by repeating the experiment and comparing the results. Bendat and Piersol (1986), Bevington and Robinson (1992), and Taylor (1982). have lengthy treatments of random error analysis as well as additional definitions of random data.

Procedures for Analyzing Data

Procedures for automatically analyzing data can be divided into two primary categories: procedures for analyzing individual records, and procedures for analyzing a collection of records (Bendat and Piersol 1986). Some individual record procedures which we have found helpful include simple static range checks, and dynamic range checks or comparisons to expected values. Procedures for analyzing collections of records are considerably more complex and typically involve measurements of central tendency and dispersion, measurements of periodicity, and frequency or spectral analysis. At a minimum, all records should be checked against static range checks.

In the LoanSTAR program all channels are run through static high-low checks using the ARCHIVE program and are visually inspected each week using a common summary plot to facilitate comparison to previous weeks.

What About Missing Data?

Missing data can present problems. Often, when one finds individual or groups of records that do not meet prescribed error criteria, the simplest thing to do is to declare it to be "missing data" and replace those records with the appropriate indicator (we prefer to use -99). Although this is the safest way to assure that the data base is not contaminated with bad data, the practice can present problems during the analysis phase.

In general, missing data can be dealt with by: 1) throwing it out, 2) embedding a replacement value, 3) interpolation, and 4) replacing it with a synthesized or calculated value. Regardless of which method is chosen, always keep the raw data and the instructions necessary for recreating any embedded or estimated values.

2.7 SUMMARY

This section has attempted to provide a review of measurement techniques, including a review of electricity monitoring, temperature measurements, humidity measurements, and flow and Btu measurements. Helpful hints from experiences gained in the LoanSTAR program have also been provided concerning the installation and calibration of sensors, and the analysis of errors.

TABLE 2-1 *Summary of Flow Meter Characteristics.* This table presents a summary of flow meter characteristics which has been assembled from experiences gained in the LoanSTAR monitoring program and other useful sources.

FLOWMETER	DIRTY	CLEAN	RANGABILITY	LOW FLOW	PURCHASE COSTS ($)	INSTALL COST ($)	ACCURACY UNCALIBRATED (INCLUDING TRANSMITTER) (note 1)	APPLICATIONS
ORIFICE PLATE	A	G	THE RANGE OF ALL THE DIFFERENTIAL METERS IS LIMITED BY THE PRESSURE TRANSDUCER IN USE	G	$800	$1500	± 1-2 % FULL RANGE SCALE (FRS)	GOOD FOR LOW FLOWS IN CLEAN WATER
VENTURI TUBE	G	G		G	$1500	$1500	± 1-2% FRS	GOOD FOR CONTROLS & MONITORING IF PRESENT
NOZZLE	G	G		G	$2000	$2000	± 1-2% FRS	GOOD FOR STEAM AND HIGH PRESSURE GASSES
PITOT TUBE	P	G		G	$500	$500	± 5% FRS	GOOD FOR SPOT CHECKS IN CLEAN FLUID
ANNUBAR	P	G		G	$1500	$500	± 1% OF RATE	VERY GOOD FOR LOW FLOW IN CLEAN FLUID
TURBINE	P	G	1-30 FPS	P	$1500	$1500	± 1% OF RATE	VERY GOOD METER IN CLEAN FLOW
VORTEX	G	G	1-30 FPS	G	$3500	$1500	± .5-1.5% OF RATE	GOOD MULTI PURPOSE METER
TANGENTIAL PADDLEWHEEL	A	G	1-30 FPS	P	$500	$500	± 2.5% OF RATE	GOOD FOR MONITORING
INSERTION TURBINE	P	G	1-30 FPS	P	$1500	$500	± 1% OF RATE	GOOD FOR MONITORING
TARGET	A	G	1-30 FPS	P	$1500	$500	± 1.5-5% FRS	GOOD FOR MONITORING
ULTRASONIC TIME OF FLIGHT DOPPLER	G G	G G	.5-30 FPS .5-30 FPS	A A	$2000-$3500	$500	± 5 %FRS ± 5% FRS	GOOD FOR SPOT CHECKS
MAGNETIC	G	G	.5-30 FPS	G	$3000	$1500	± 1% FRS	GOOD METER IF BUDGET IS NOT A FACTOR
MASS FLOW	A	G	.5-30 FPS	G	$3500	$1500	± .2-1% FRS	GOOD IN LOW CLEAN FLOW AND SMALLER PIPES

G - GOOD A - AVERAGE P - POOR

NOTE: 1. VALUES FROM MILLER 1989.

2. ARE OTHER VALUES IN THIS TABLE ARE APPROXIMATE AND ARE GIVEN AS REFERENCE VALUES C PRICES AND CHARACTERISTICS WILL CHANGE AS CONDITIONS DICTATE

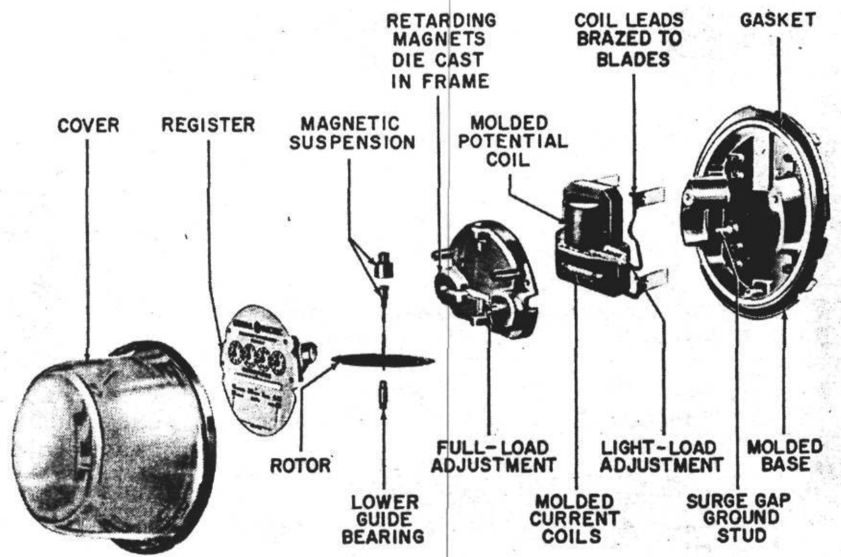
TABLE 2-2 List of Potential Sensors for Calibration. This table provides a list of possible sensors that will need to be calibrated in a large monitoring program (Turner et al. 1992).

Liquid Flow	Temperature	Hydraulic Pressure	Air Pressure
1. Orifice 2. Venturi 3. Flow Nozzle 4. Turbine 5. Vortex 6. Magnetic 7. Ultrasonic 8. Paddlewheel 9. Impact 10. Mass - Coriolis 11. Mass - Thermal 12. Transducers kJ or BTU	1. Thermometer 2. Thermocouple 3. RTD 4. Temperature portion of Humidity sensor 5. Thermistor 6. I.C. Temperature Sensor	1. Pressure Gauge 2. Pressure Transducer	1. Pressure Gauge 2. Manometer 3. Pressure Transducer
Humidity	**Electrical Power**	**Electrical Current**	**Electrical Voltage**
1. Dew/Frost Point Sensor 2. Psychrometer 3. Thin Film Polymer 4. Mechanical (Dimensional) 5. Dielectric Crystal	1. Watt-hour meters 2. Watt transducers	1. Clamp-on Amp Meter 2. Current Transformer	1. Voltmeter 2. Multimeter 3. Potential Transformer
Air Flow	**Rotational Speed**	**Solar Radiation**	**Illumination Levels**
1. Pitot Tube 2. Hot-Wire Thermo-anemometer 3. Rotary Device 4. Flow nozzles	1. Contact Tachometers 2. Non-contact Tachometers 3. Strobes 4. Reflective Tachometer	1. Pyranometer	1. Light Meter

TABLE 2-3 Ranges and Accuracies of the Calibration Facility. This table provides a summary of the intended range and accuracies of each of the calibration stations at the facility (Turner et al. 1992).

Liquid Flow	Temperature	Hydraulic Pressure	Air Pressure
Range: 5 to 950 GPM 0.3 to 59.9 l/s	Range: -40 to 500 F -40 to 260 C	Range: 0 to 500 PSI 0 to 3.5 MPa	Range: 0 to ±24 "WG 0 to ±609 mm WG
Accuracy: ± 1% of reading	Accuracy: ±0.4 °F ±0.2 °C	Accuracy: ± 0.25% Reading	Accuracy: ±0.01 "WG ±0.25 mm WG
Humidity	**Electrical Power**	**Electrical Current**	**Electrical Voltage**
Range: 10 to 95% RH 32 to 125 °F 0 to 52 °C	Range: 0 to 100 Amps	Range: 0 to 40 KW	Range: 0 to 600 VAC 0 to 600 VDC
Accuracy: ± 1.5% RH, ±1.0 F, ± 0.65C	Accuracy: ± 0.5% Full Scale	Accuracy: ± 0.5% Full Scale	Accuracy: ± 0.5% Full Scale
Air Flow	**Rotational Speed**	**Solar Radiation**	**Illumination Levels**
Range: 0 to 8000 ft/m 0 to 46.7 m/s	Range: 0 to 3600 RPM	Range: 0 to 500 Btu/ft^2 0 to 1.5 kW/m^2	Range: 0 to 1000 FC
Accuracy: ± 10 ft/m ± .05 m/s	Accuracy: ± 2 RPM	Accuracy: ± 2% Reading	Accuracy: ± 1 % Reading

FIGURE 2-1 *Basic Parts of a Watt-hour Meter*. This figure shows the basic parts of a Watt-hour meter in an exploded view (Reproduced with permission: Edison Electric Insitute's Handbook for Electricity Metering 1981).



Courtesy General Electric Co.

FIGURE 2-2 *Voltage and Current Flux Waveforms in a Watt-hour Meter*. This figure shows the basic waveforms for voltage and current flux in a Watt-hour meter and their net effect on the rotor (Reproduced with permission: Edison Electric Insitute's Handbook for Electricity Metering 1981).

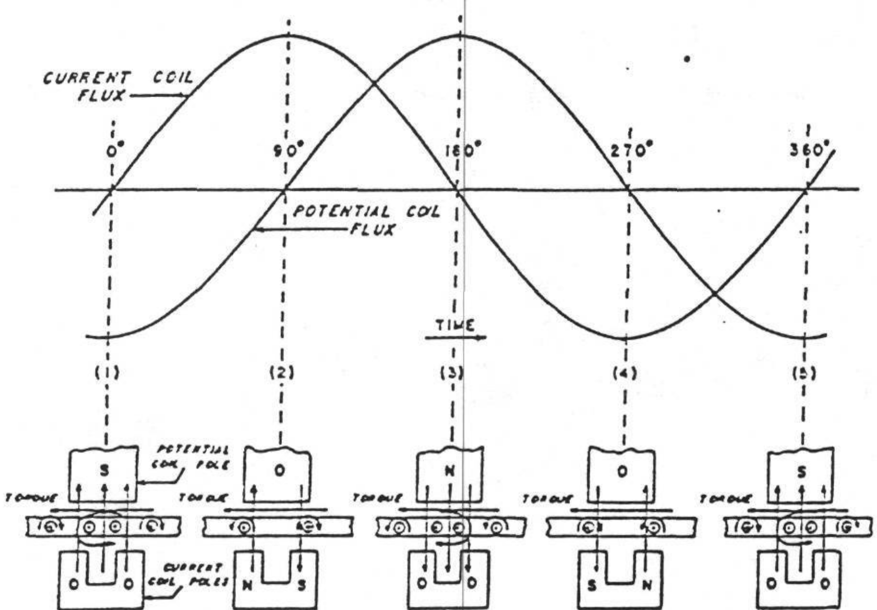


***FIGURE 2-3** Contact Mechanical Pulse Initiator for 2-wire or 3-wire KYZ pulse* (Reproduced with permission: Edison Electric Insitute's Handbook for Electricity Metering 1981).

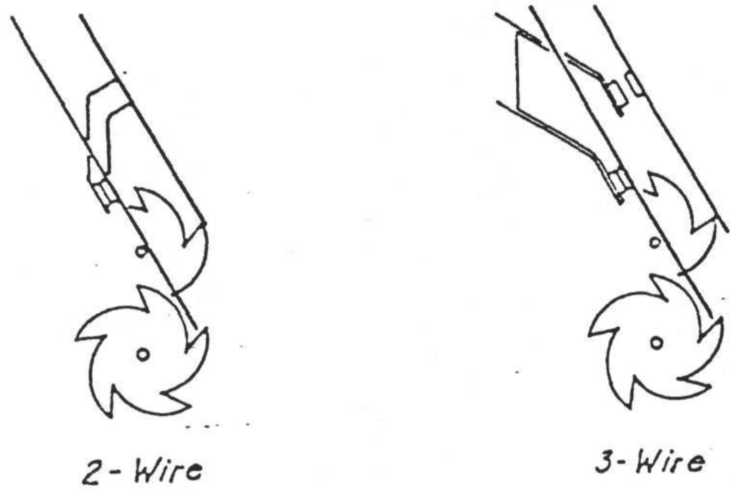


***FIGURE 2-4** Simplified Diagrams Illustrating Basic Methods of Telemetering kiloWatt-hours*. (Reproduced with permission: Edison Electric Insitute's Handbook for Electricity Metering 1981).

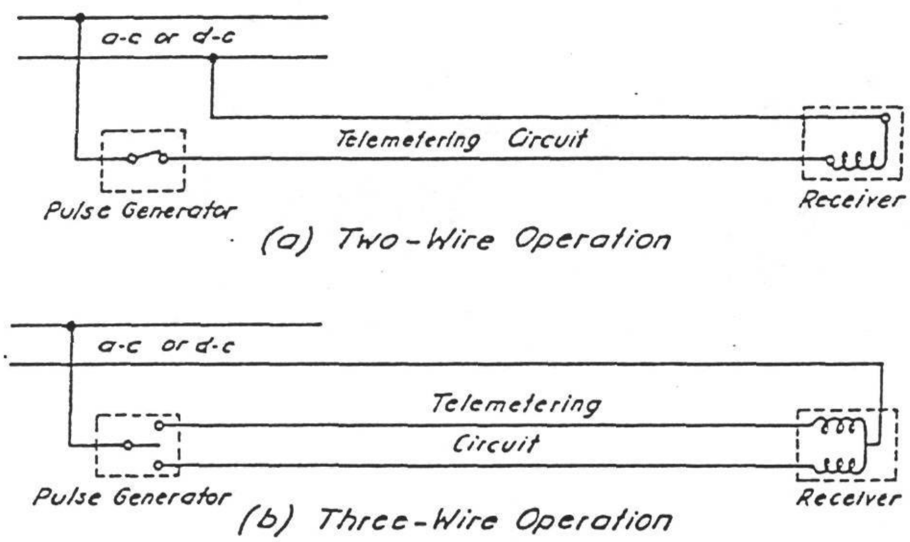


FIGURE 2-5 *Thermal Demand Meter and Circuits.* (Reproduced with permission: Edison Electric Insitute's Handbook for Electricity Metering 1981).

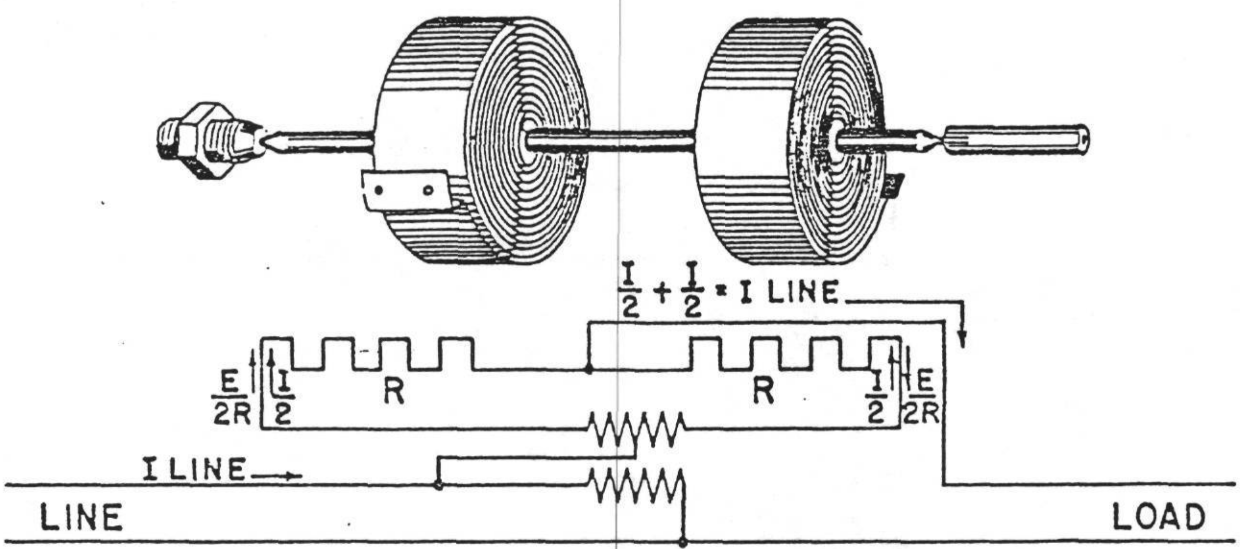
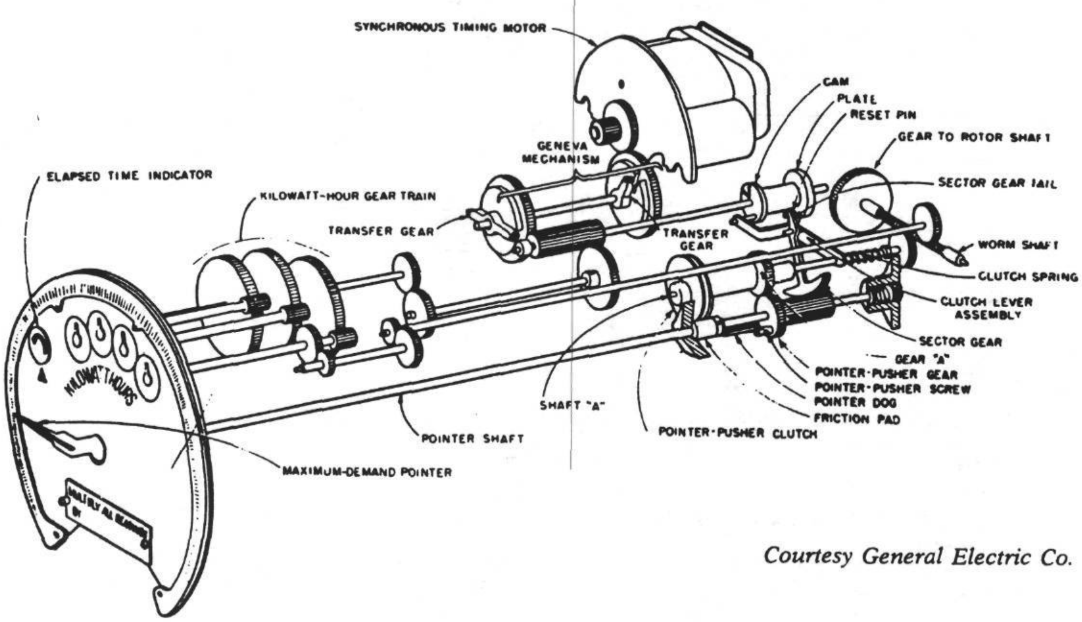


FIGURE 2-6 *Mechanical Cumulative Watt-hour Demand Meter* (Reproduced with permission: Edison Electric Insitute's Handbook for Electricity Metering 1981).



Courtesy General Electric Co.

***FIGURE 2-7** Functional Block Diagram for a Watt/Watt-hour Transducer* (Reproduced with permission: Edison Electric Insitute's Handbook for Electricity Metering 1981).

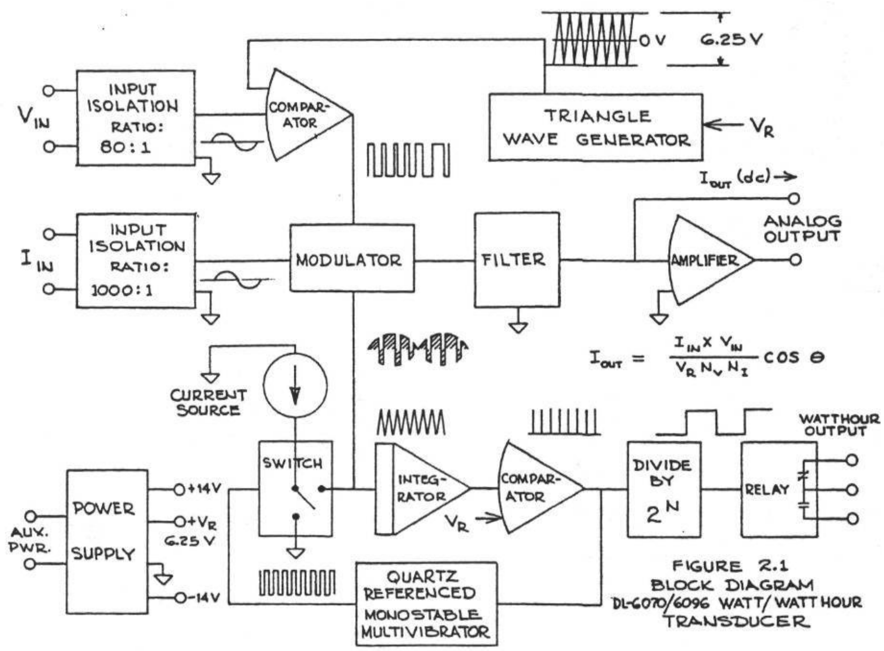


***FIGURE 2-8** Electronic Multiplier Waveforms for a Watt/Watt-hour Transducer* (Reproduced with permission: Edison Electric Insitute's Handbook for Electricity Metering 1981).

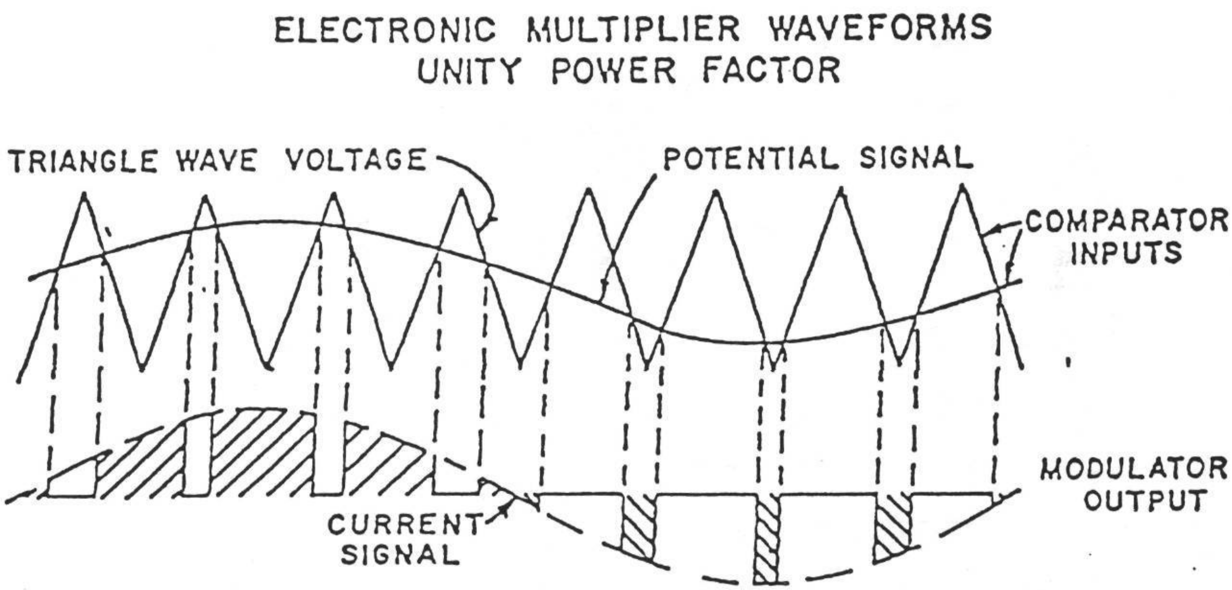


***FIGURE 2-9** Integrated, solid state Watt-hour Meter.* This figure shows a schematic of an integrated solid state Watt/Watt-hour meter of the kind used in the Synergistic recorder (Source: Schuster 1985).

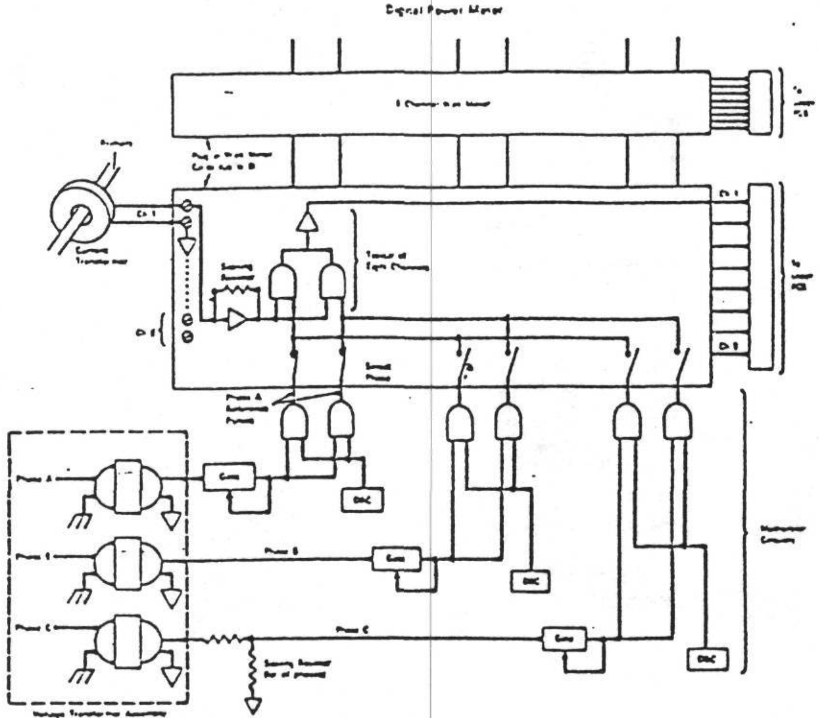


***FIGURE 2-10** Thermocouple temperature/voltage curves.* (Reproduced with permission: Doebelin 1990). This figure shows the voltage-temperature relationships for Type E, J, K, T, R, and S thermocouples.

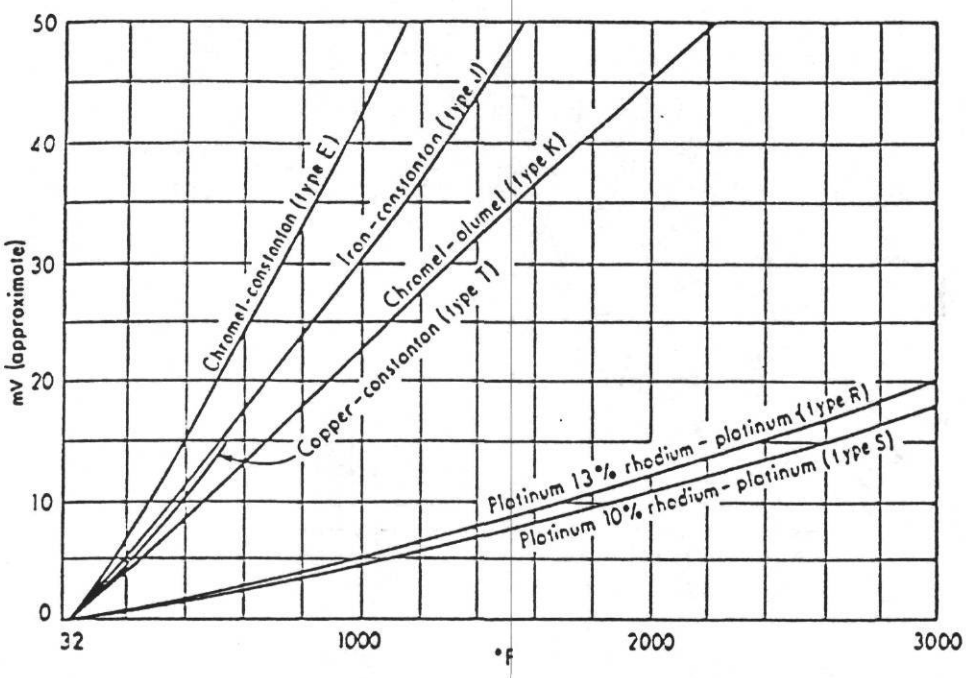


***FIGURE 2-11** Isothermal thermocouple reference junction* (Reproduced with permission: Doebelin 1990). This sketch shows an isothermal reference-junction that is commonly used in computerized thermocouple measurements.

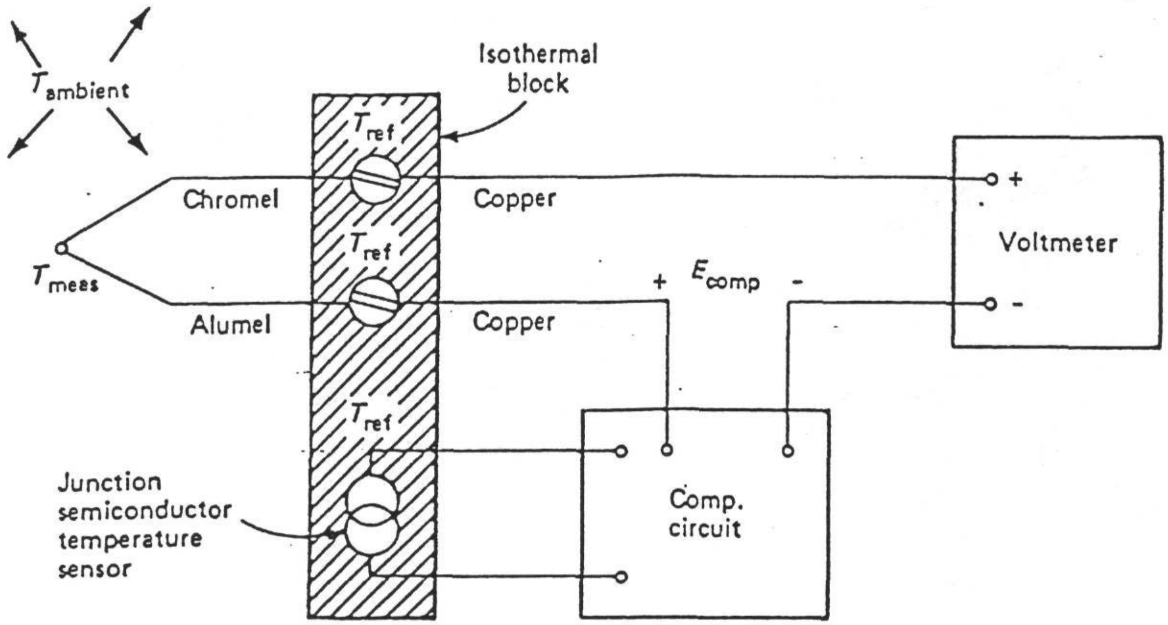


***FIGURE 2-12** Resistance-temperature curves for nickel, copper, platinum and manganin* (Reproduced with permission: Doebelin 1990).

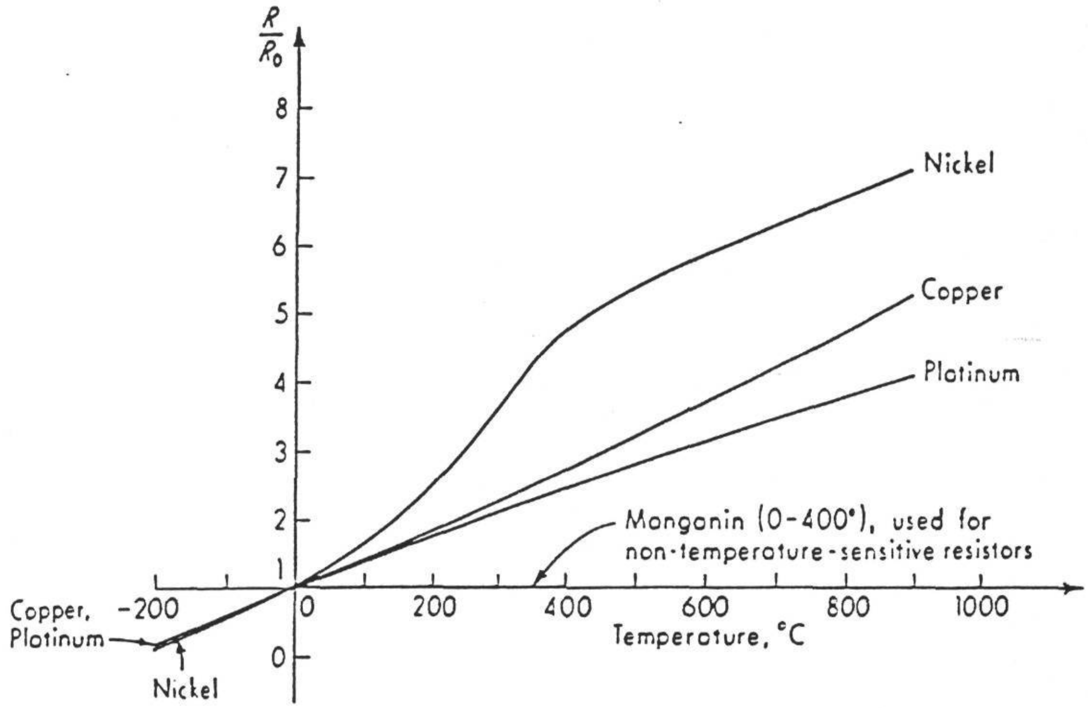


FIGURE 2-13 *Bridge circuits for Resistance Temperature Detectors - RTDs* - (Reproduced with permission: Doebelin 1990).

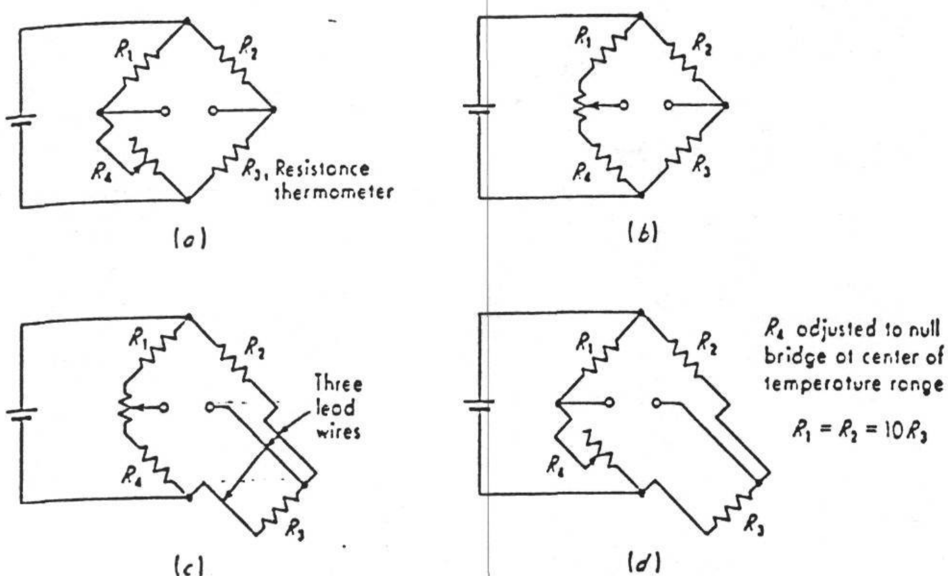


FIGURE 2-14 *Thermistor temperature-resistance curves* (Reproduced with permission: Doebelin 1990).

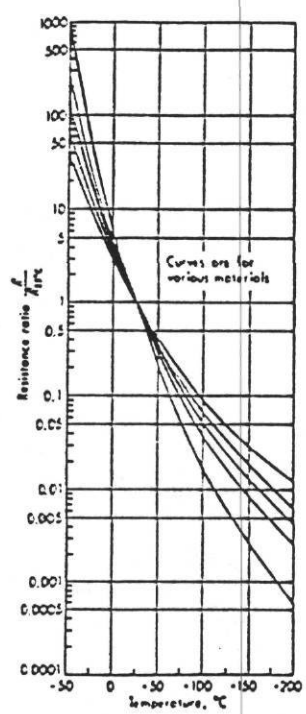
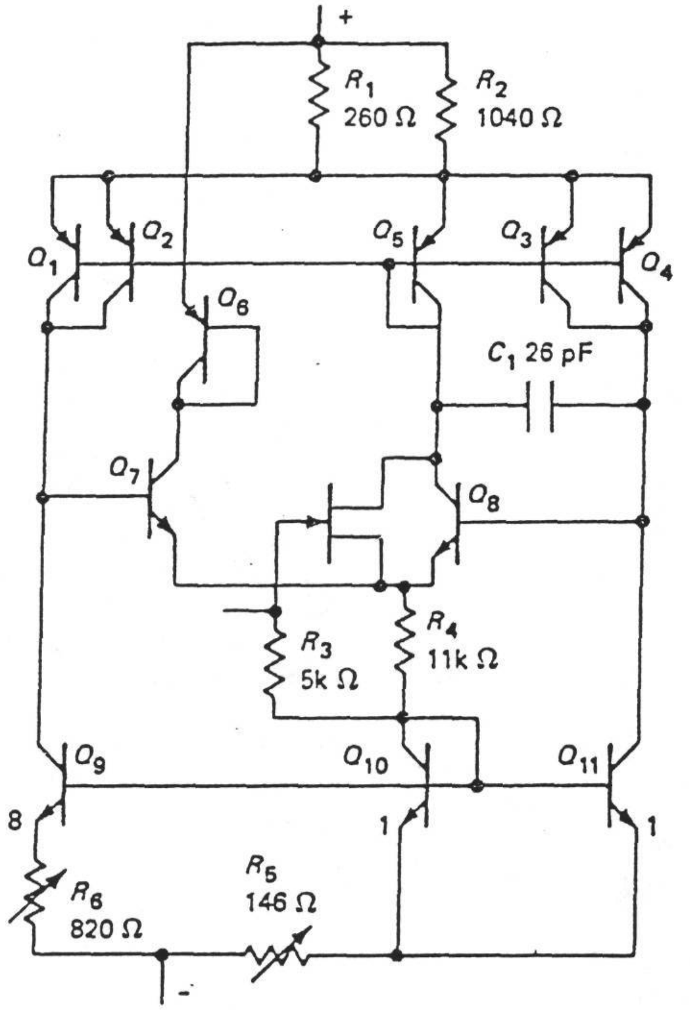


FIGURE 2-15 *Junction semiconductor temperature sensor* (Reproduced with permission: Doebelin 1990).



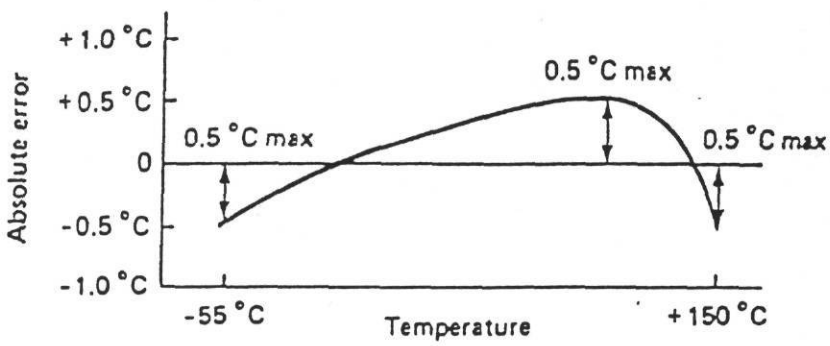


FIGURE 2-16 *ASHRAE psychrometric chart* (Source: ASHRAE).

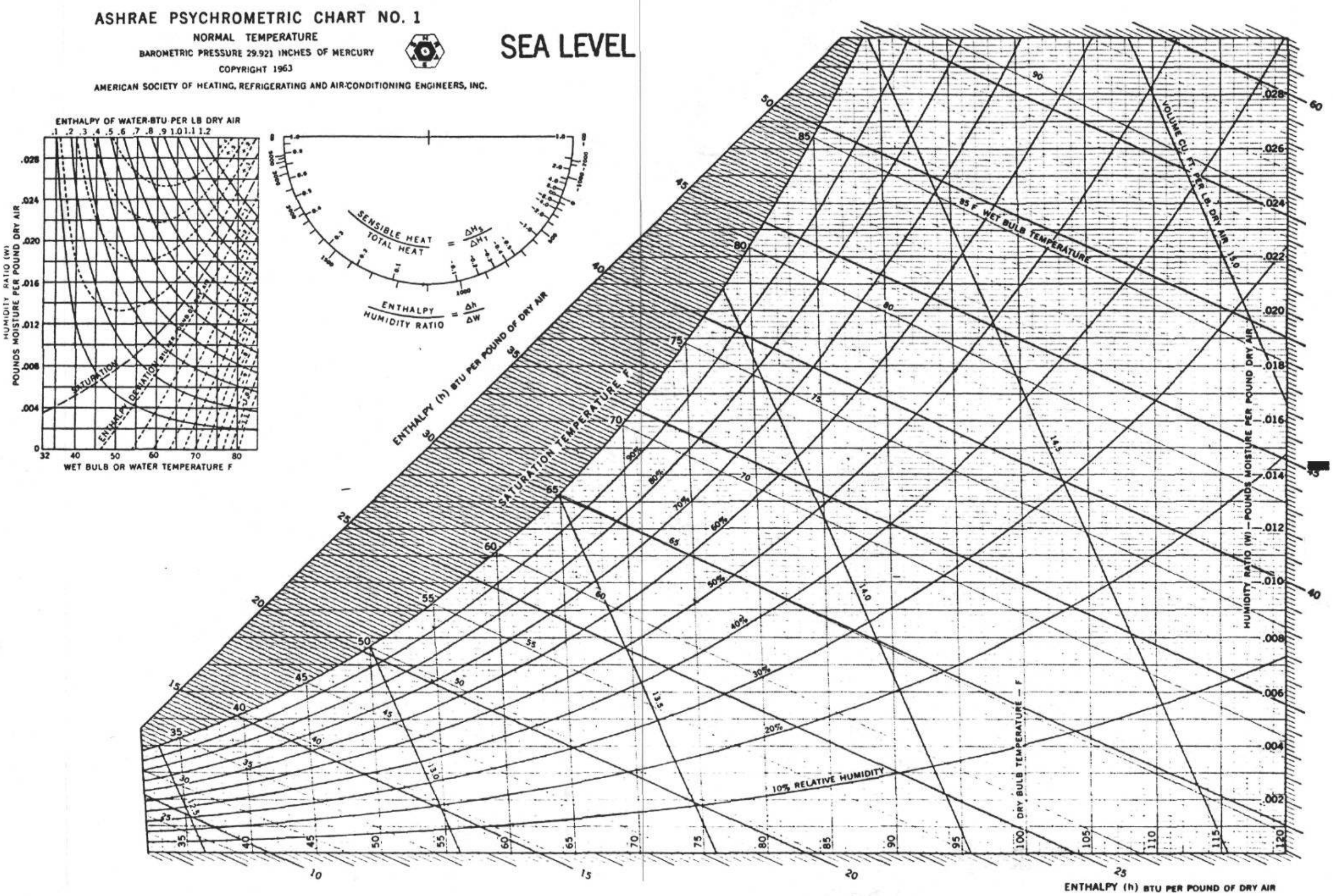


CHART 1

***FIGURE 2-17** Typical hand-held psychrometer* (Source: ASHRAE).

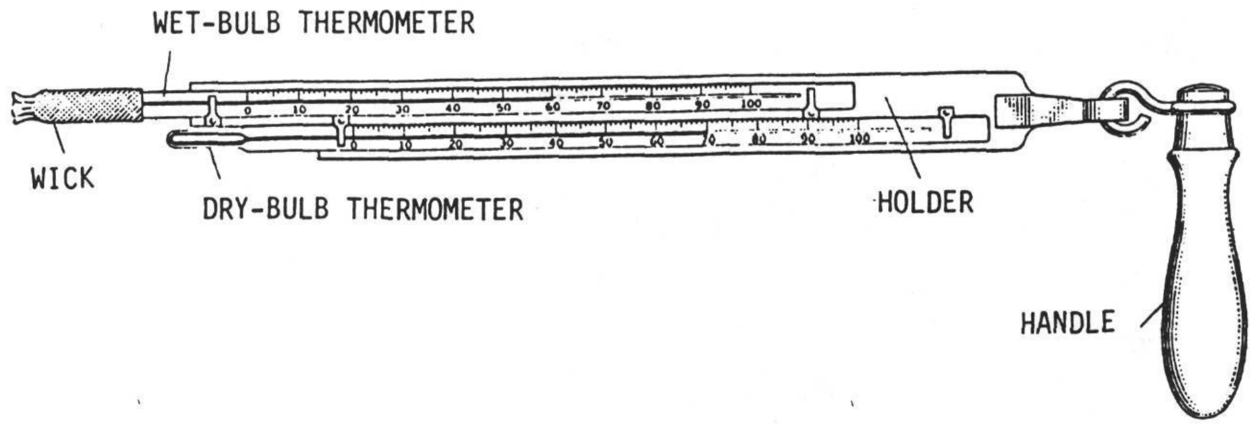


***FIGURE 2-18** Simplified schematic of a chilled-mirror dewpoint sensor* (Source: Wiesman 1989).

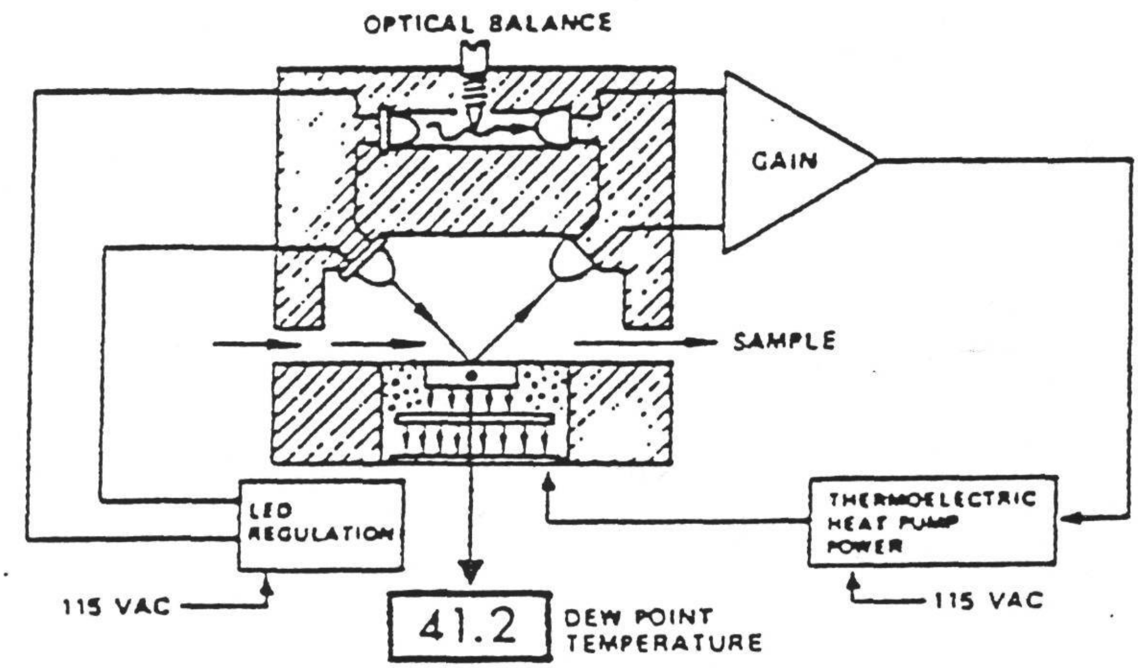


***FIGURE 2-19** A Pope-type ion-exchange resin sensor* (Source: Hurley 1985).

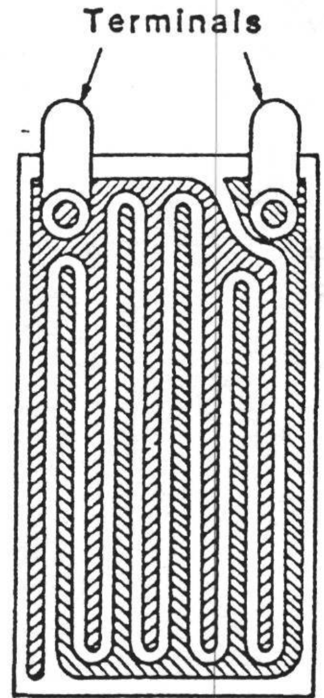


***FIGURE 2-20** An aluminum oxide capacitance-type sensor or Jason hygrometer* (Source: Hurley 1985).

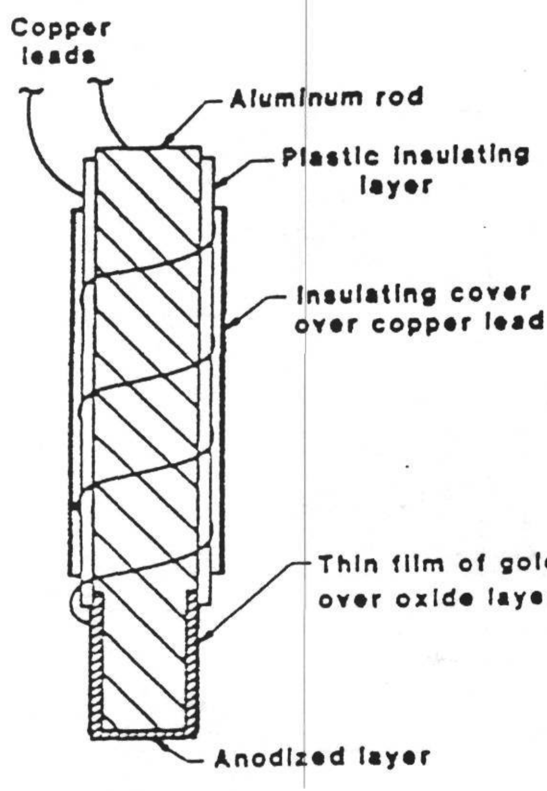


FIGURE 2-21 Diagram of a thin-film polymer-type capacitance sensor (Source: Hurley 1985).

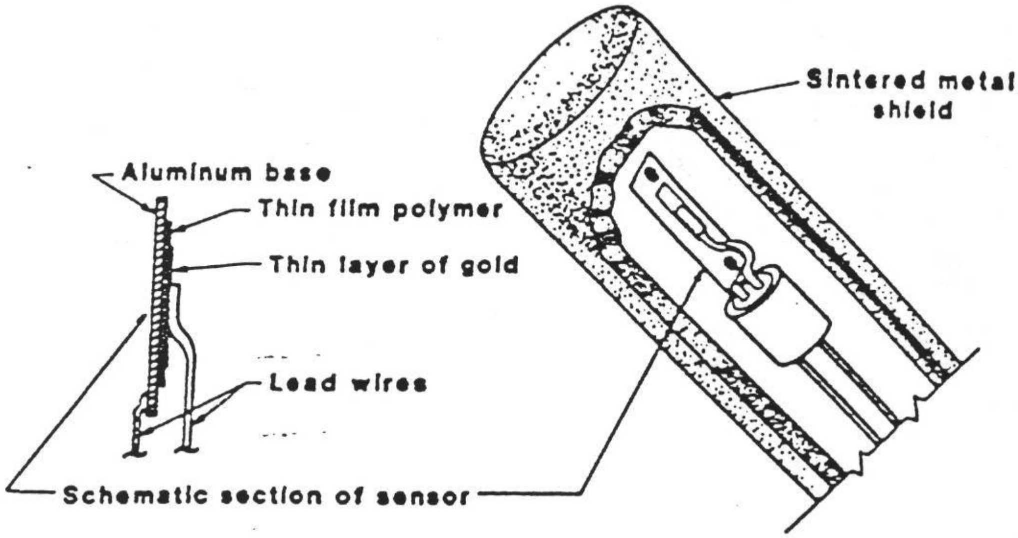


FIGURE 2-22 Orifice plate meter (Reproduced with permission: Doebelin 1990).

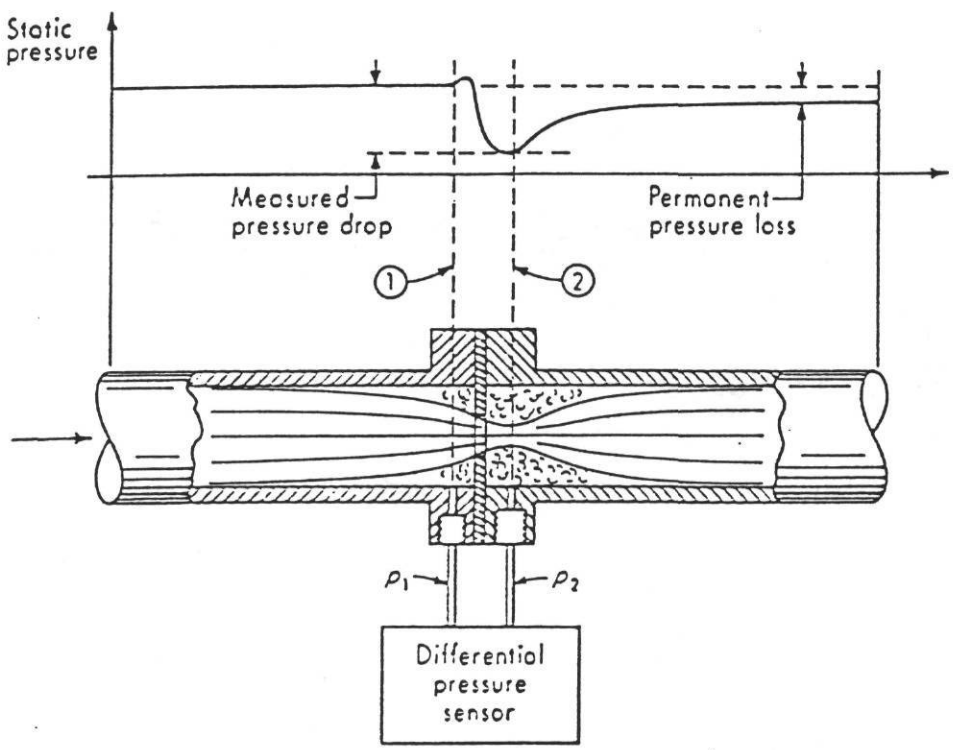


FIGURE 2-23 Venturi meter (Source: Baker and Hurley 1984).

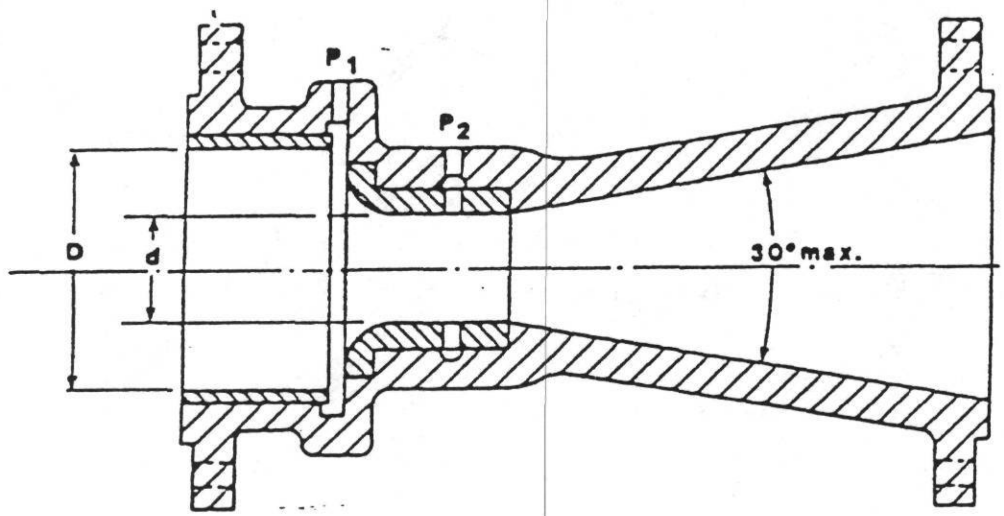


FIGURE 2-24 Pitot tube meter (Source: Baker and Hurley 1984).

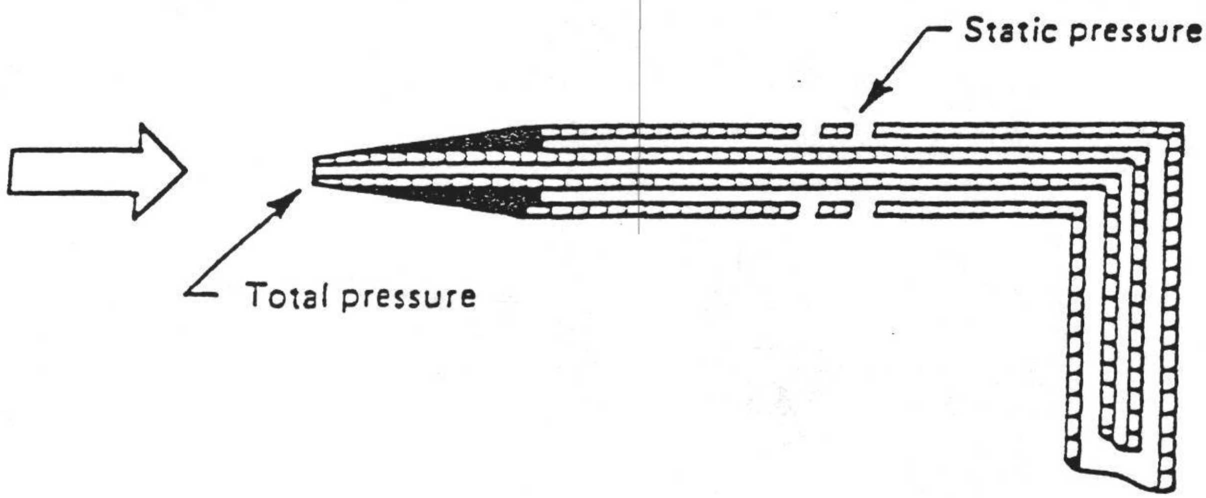


FIGURE 2-25 Averaging-type pitot meter (Reproduced with permission: Doebelin 1990).

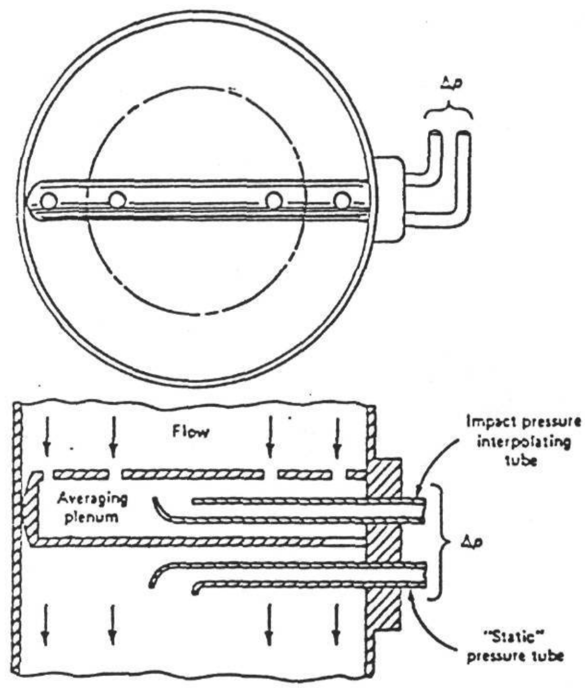


FIGURE 2-26 Variable-area meter (Reproduced with permission: Miller 1989).

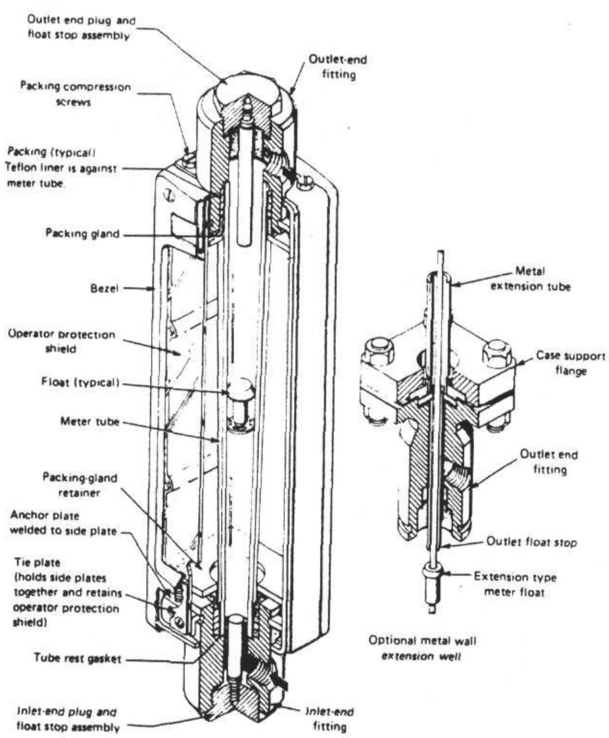


FIGURE 2-27 *Positive displacement meter* (Source: Hurley 1985).

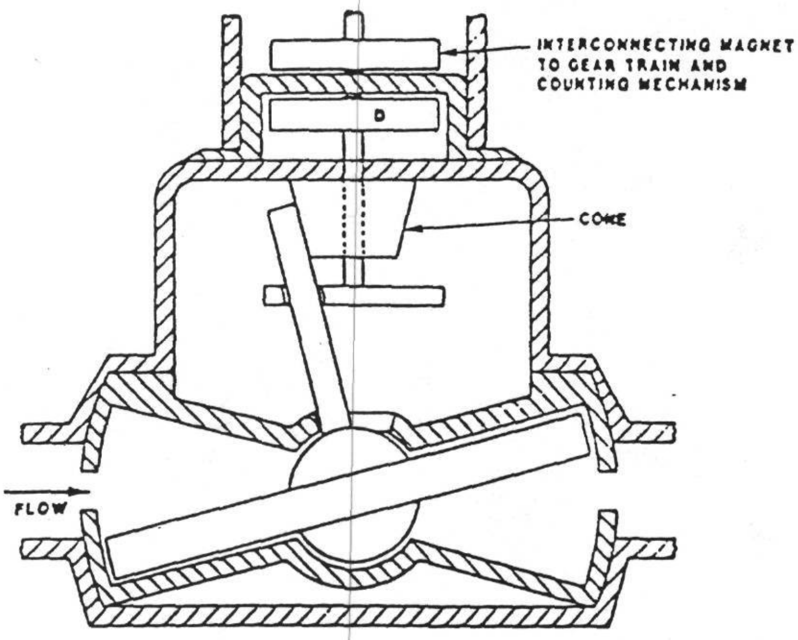


FIGURE 2-28 *Turbine meter* (Source: Hurley 1985).

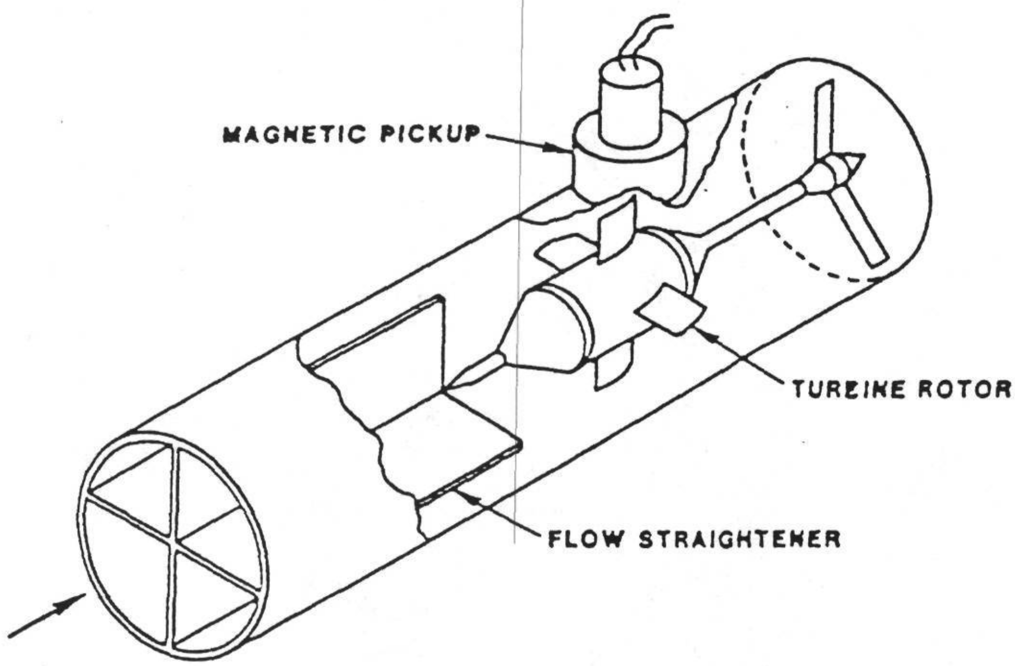


***FIGURE 2-29** Tangential paddlewheel meter.*

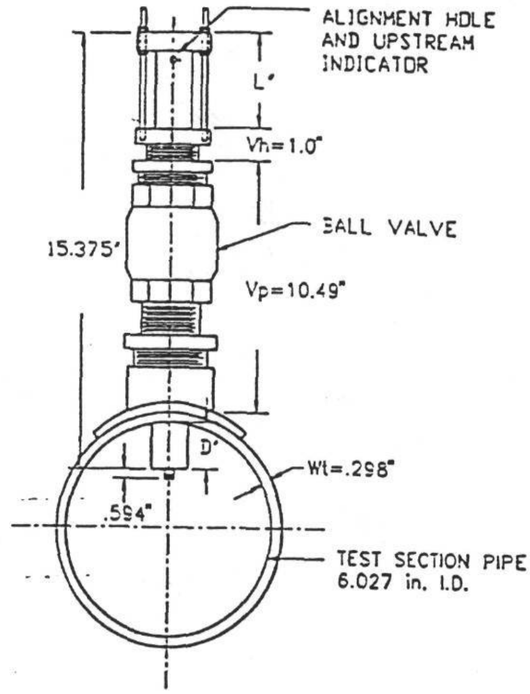


***FIGURE 2-30** Target meter* (Source: Hurley 1985).

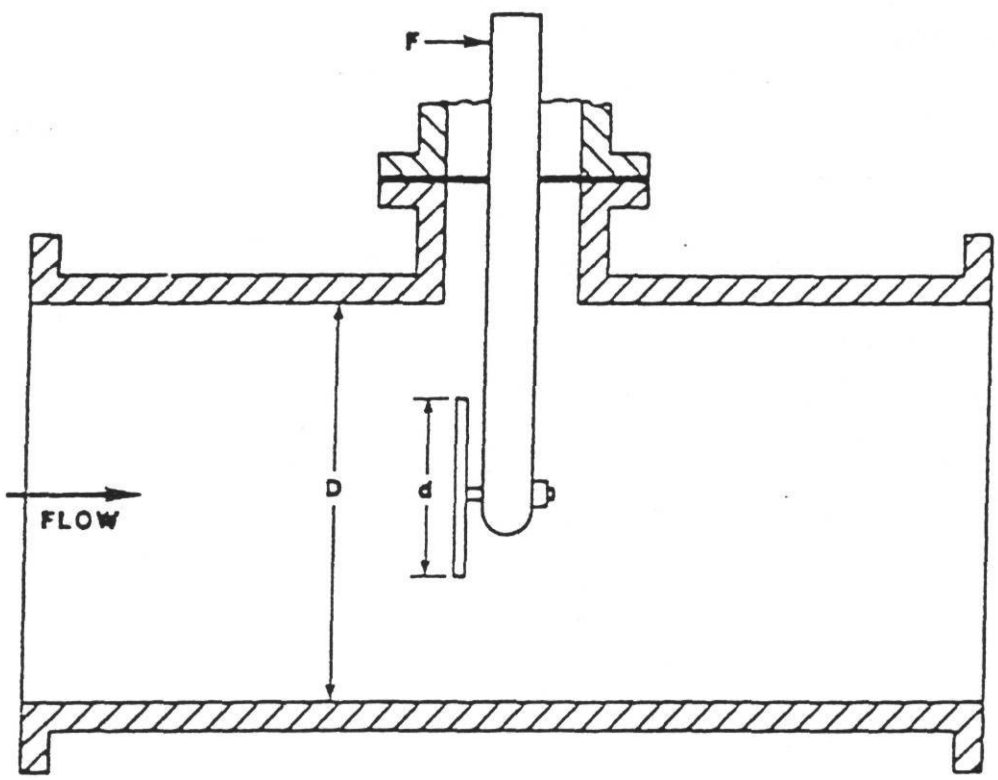


FIGURE 2-31 Vortex meter (Reproduced with permission: Doebelin 1990).

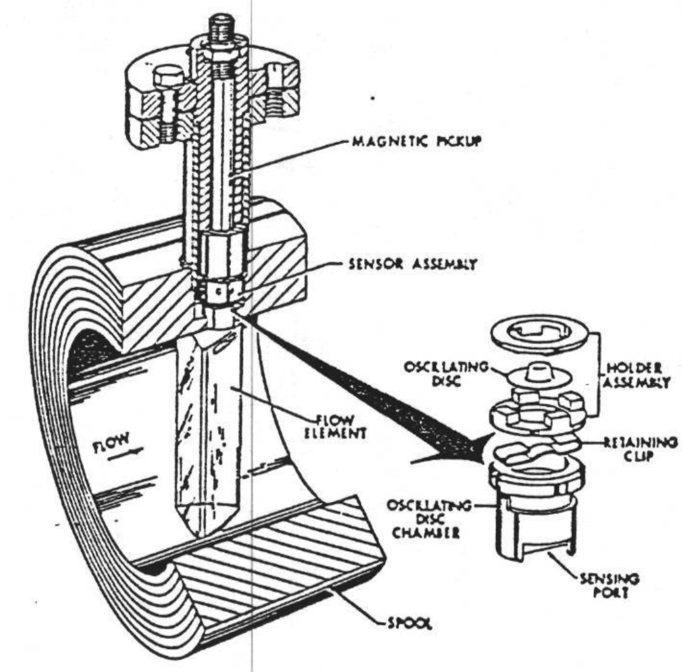


FIGURE 2-32 Coriolis mass flow meter (Reproduced with permission: Miller 1989).

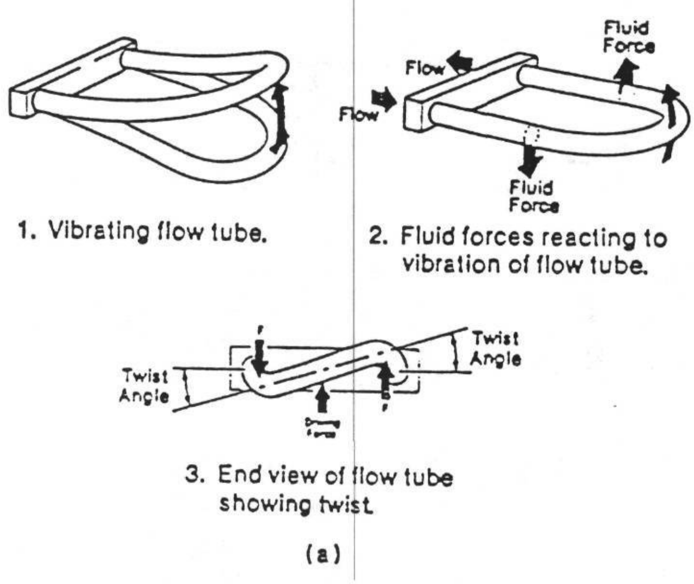


FIGURE 2-33 *Angular momentum mass flow meter* (Reproduced with permission: Doebelin 1990).

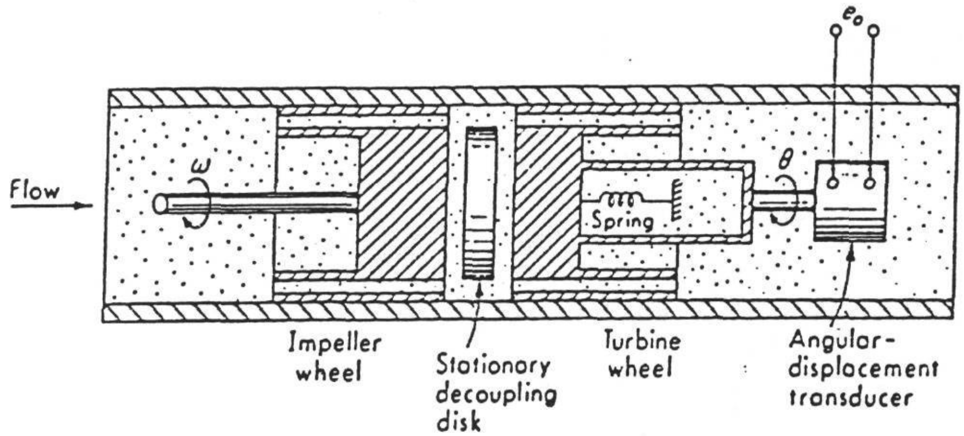


FIGURE 2-34 *Ultrasonic meter* (Source: Hurley 1985).

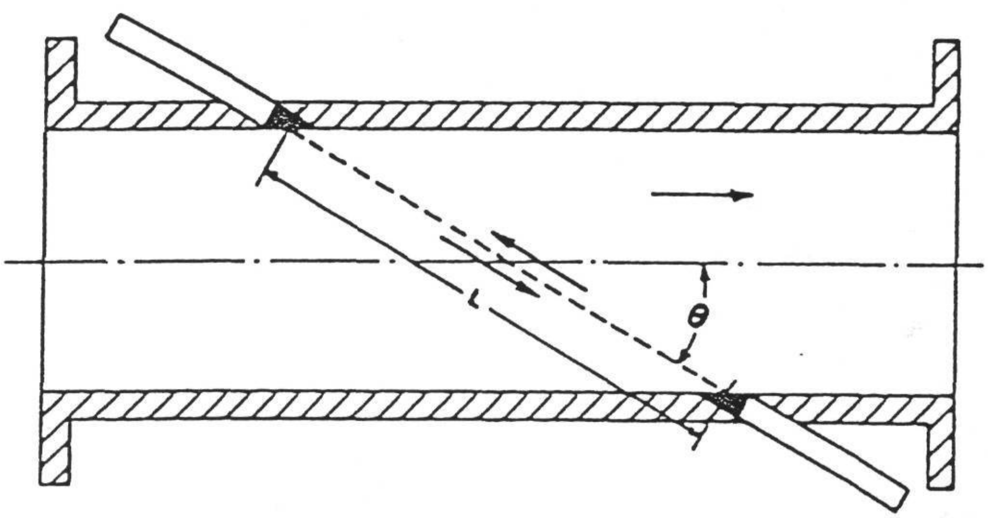


FIGURE 2-35 *Magnetic meter* (Reproduced with permission: Doebelin 1990).

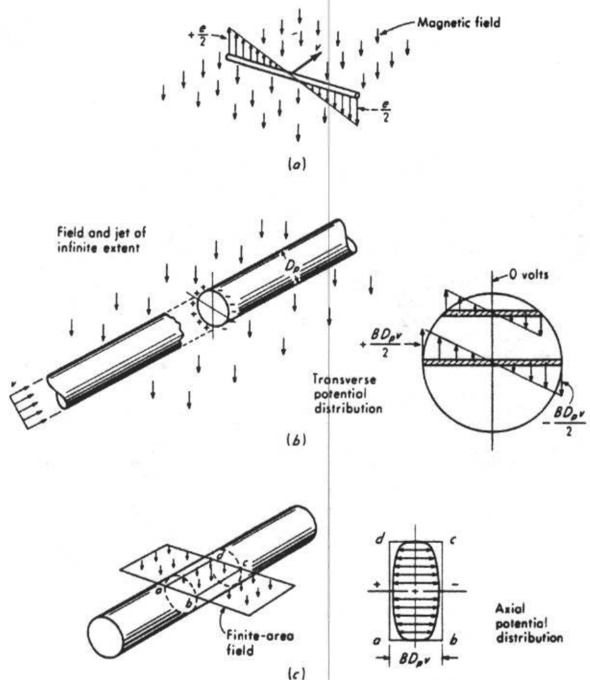


FIGURE 2-36 *Mechanical totalizing Btu meter* (Reproduced with permission: Doebelin 1990).

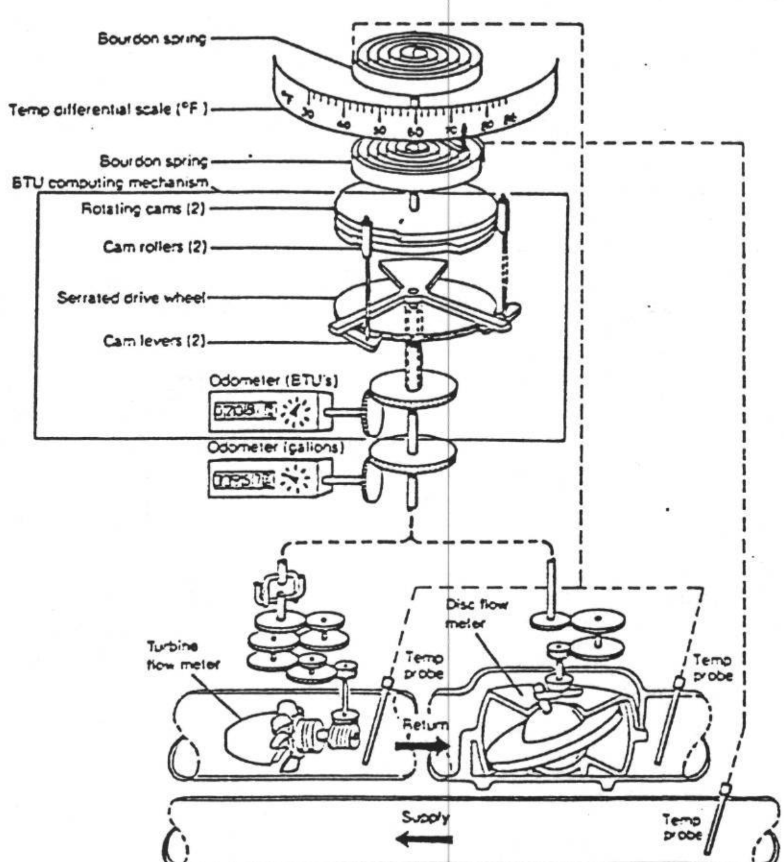


FIGURE 2-37 *Photo of the LoanSTAR flow meter test facility.* This figure shows the LoanSTAR dynamic-weight flow meter test facility at Texas A&M (Claridge et al. 1991).

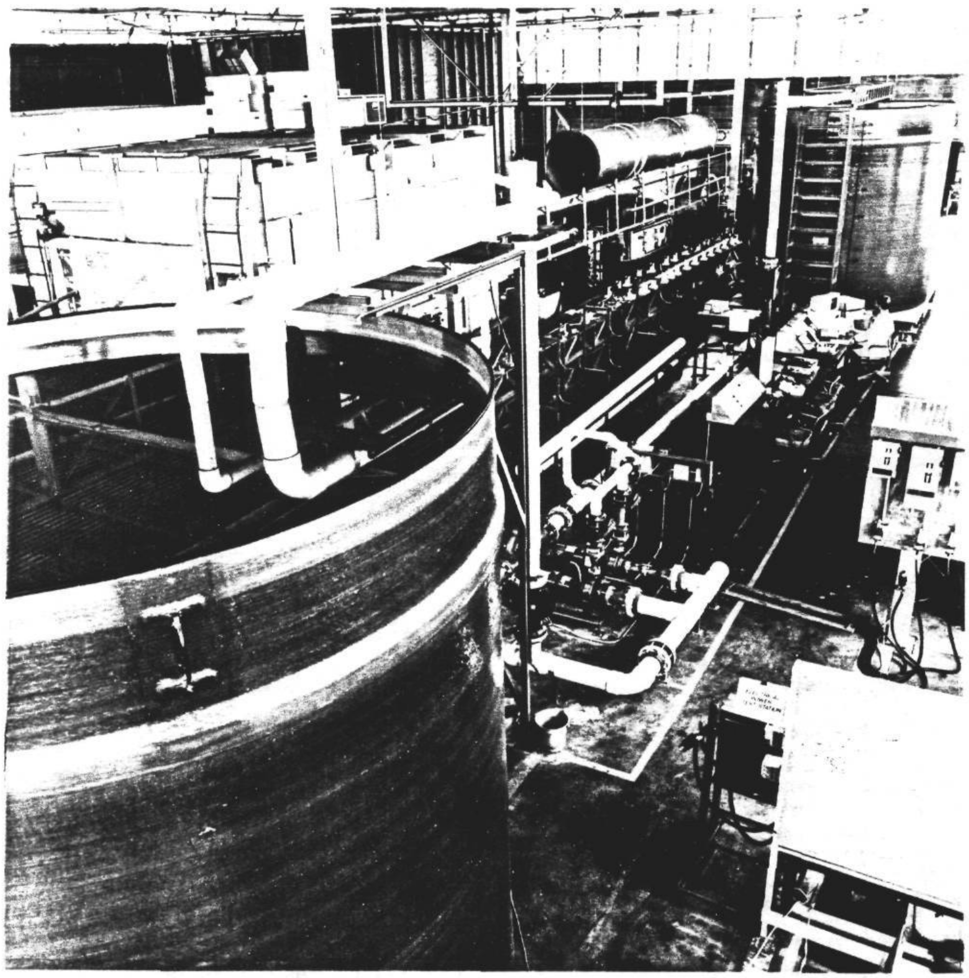


FIGURE 2-38 *Results of different flow meters in a 6" pipe.* This figure shows the results of tests that were performed on magnetic and non-magnetic flow meters in a 6" pipe for velocities varying from 1/2 to 10 feet per second (Robinson 1992).

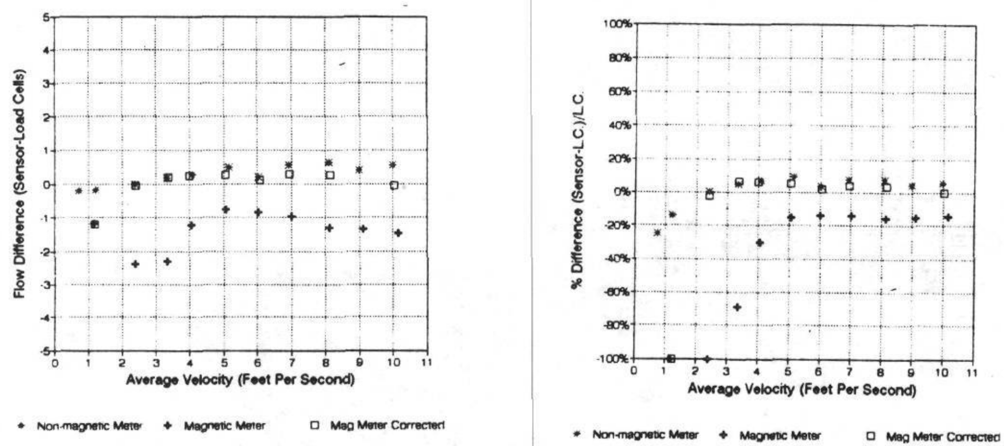
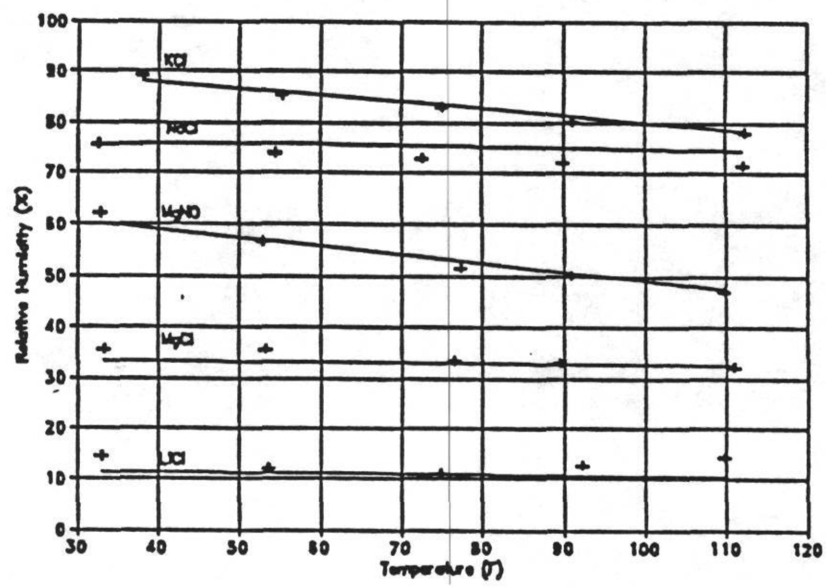


FIGURE 2-39 *Typical results from the calibration of humidity sensors using the NIST saturated salt solutions* (Bryant and O'Neal 1992).



3.0 USING A DATA LOGGER

This section that follows is intended to be a "survival guide" to instruct the first time user in setting-up and using the data logger that was developed by Battelle/PNL for the USDOE and is now commercially available (Synergistics 1990). This logger is the primary logger in the Texas LoanSTAR program. Other data logger survival guides (and software translators) are available from Texas A&M. Additional details may be necessary to connect the logger to different configurations. The reader is referred to the manufacturer's manual for additional details. A descriptive paper by Schuster (1985) has been included in the appendix to this workbook to add further details concerning the basics behind the logger.

3.1 CONNECTING THE SENSORS TO THE LOGGER.

Figure 3-1 is a generalized diagram that shows the logger in an installation where power is being monitored from 3 circuits in a 3-phase, 4-wire electrical panel. In this diagram solid core, shunted current transformers (CTs) are shown installed around circuits leading to three electrical loads. The potential transformer (PT) is shown connected to voltage leads from A, B, C, neutral and ground. A 24 VAC transformer has also been installed (in the electrical panel) to provide the basic power for the logger. The phone line connection (not shown) would also require a connection.

Configuration of the Logger.

Figures 3-2 and 3-3 show the basic termination boards for the logger. In Figure 3-2 the termination board for the digital (Dxx), power (CTxx), voltage (L2xx), and power supply (24xx) points are shown. Figure 3-3 shows the termination board for the analog channels. In this particular logger different combinations of voltage (Vxx), current (Axx), and ground (GND) channels must be used to connect purely resistive or 4-20 mA loads. In this logger non-resistive (4-20mA) signals are connected through the use of resistive "headers" that must be plugged in for each circuit.

Connecting a digital pulse signal. Figure 3-4 gives an illustration of a digital signal connection to the logger. The on/off pulse signal can be that provided by a 2-wire KYZ pulse (which will need field scaling), or any other dry contact telemetering circuit. In Figure 3-4 the 2-wire unshielded leads for a digital channel are shown connected to digital channel "0" on the main termination board.

Connecting a 2-wire resistive RTD signal. Figure 3-5 shows how a resistive 2-wire RTD could be connected to the logger. A resistive load can be connected directly to the logger without the need for a header. The shield wire is grounded only to the logger to avoid a ground loop. NOTE: The use of a 2-wire RTD usually limits the length of the analog signal because of the resistance that is added by the intervening wire. Any 2-wire RTD should be field calibrated (or calibrated in a laboratory) with the exact length of lead wire to assure that the extra resistance from signal wire is accounted for in software.

Connecting a 2-wire, 4-20 mA signal with the use of a 200 Ohm resistive header. The basic diagram for connecting a 2-wire 4-20 mA current loop is shown in Figure 3-6. Connecting a 4-20 mA current loop to the logger requires that a resistive header be inserted for this channel. The header basically provides a 200 Ohm resistor across the Ax to GND terminals.

Connecting multiple CTs using a summing module to one power channel. Electrical power, voltage, current, and power factor are measured by the on-board solid state Watt/Watt-hour transducers. The primary input(s) that are needed for this are properly sized shunted current transformers (CTs) and a potential transformer (PT). In some cases where high voltage loads (+480 VAC) are being monitored, a combination of PTs may be required. In Figure 3-7 the basic connection for a single power signal is shown being connected to channel "0" after it has been totalized in the summing module. The use of such summing modules is convenient when one has multiple CTs (of the same rating) that need to be accumulated into one signal. Care must be taken to assure that the circuits being monitored are all on the same phase and are of the same size for this to work properly. Any CT should be field checked with hand-held "clamp-on" ammeter. All CTs must be installed with the same polarity. This is accomplished by aligning the arrows, or dots, toward the electrical source.

Connecting the PT to provide a voltage signal. In order to provide the power measurements a potential transducer must be connected to each 3-phase feed being monitored. This logger provides PT inputs for two 3-phase feeds. Additional feeds can be accommodated with additional loggers. NOTE: Care must be taken to align the A,B,and C phases with their respective CTs both on the termination board and in software. This is the single most common mistake that is made with any logger -- incorrect configuration of power monitoring. Figure 3-8 provides an example of how the potential transformer is connected to the logger.

3.2 SURVIVAL COMMANDS FOR PROGRAMMING THE LOGGER.

The Synernet software that is provided by the manufacture with each logger is a reasonably powerful polling software package. Like any software package, there is a learning curve to using the software and the user should allow time to become familiar with the software by practicing on a site or two before committing to a monitoring project. The section that follows provides a survival guide to setting-up and polling a logger. In each example enough details are provided to illustrate the basic steps that are necessary for setting-up a site and polling a site. As such, only those SYNERNET, CONFIG, QUEUE, and PARSET commands that are necessary to accomplish this are discussed. For more information about the SCHEDULE, and MESSAGE commands the reader is referred to the Synergistics manual (Synergistics 1990).

SYNERNET

SYNERNET is the menu-driven software that is provided by the manufacturer to schedule and poll a logger. It basically contains 5 sub-programs that can be used to perform the different functions as shown in Figure 3-9. Each of the 5 sub-programs can be executed separately by typing the executable name (i.e., PARSET <CR>) to execute directly or by beginning a session with SYNERNET and working one's way down the menu tree. SYNERNET always begins a session by checking the PC's time and date.

CONFIG

The first thing that must be done with any PC after installing SYNERNET software is to setup the configuration file. This tells SYNERNET important features about the PC such as which communication port is being used, etc. This is accomplished by running the CONFIG routine. In Figure 3-10 an outline of the different questions that must be answered with CONFIG is shown. In general, one must declare whether there is a color or monochrome monitor, how long to wait (30 seconds) for a response, whether one is using RS232 or a MODEM and whether it is COM1 or COM2 on the PC, and (if necessary) what the phone numbers are for calling the site. Exiting the CONFIG menu (as is similar to most SYNERNET menus requires pressing the <F10> key to save what has just been entered on the screen.

QUEUE

The QUEUE program helps activate a logger/site for use by the SYNERNET software. QUEUE is used to define loggers and when/if those loggers are to be automatically called with the SCHEDULE software. Figure 3-11 illustrates the basic configuration of the QUEUE software. QUEUE should be used to set up a logger prior to using the PARSET program to manually poll the logger. The main advantage to using QUEUE is that any resultant list of loggers is organized by site number in QUEUE whereas in PARSET the list of loggers remains in the order that they were added. This can create a problem with large numbers of loggers. Either QUEUE or PARSET can be used to activate a logger.

PARSET

PARSET is the workhorse of the SYNERNET software system. In general, PARSET is used to add a logger to the network, configure channels in a logger, manually poll a logger, view realtime data, and/or download data if desired. Prior to running PARSET, QUEUE can be run to add a logger into a sorted list, or the logger can be added to the network in PARSET. Figure 3-12 illustrates the menu arrangement within PARSET. For the most part one must always choose a logger, then either edit the channels, or (if this has not already been accomplished) connect to a site, and use the terminal command to communicate to the logger after connecting.

3.3 SETTING-UP AND POLLING A LOGGER.

This section discusses how to set-up a logger, configure the logger, call the logger, and download data using an example as a guide to illustrate the steps. In this example, a logger has been configured as illustrated in Figure 3-13 to record:

- one channel of electrical power, kW (Channel 0, 1, 2 & 3),
- one channel of electrical power, kV (1)
- one channel of electric voltage, VOLT (2)
- one channel of the electric current, AMPS (3)
- one channel of temperature (1000 Ohm RTD, Channel 4),
- one channel of humidity (4-20 mA RH, Channel 5),
- one channel of chilled water flow, pulse GAL (Channel 6),
- one channel of chilled water energy, BTU (Channel 7).

This next section walks the reader through how the logger was set up and configured and what the data look like coming from the logger.

Setting-up and polling a logger.

Any new logger that is added to a network needs to be set up with either the QUEUE program or PARSET. In the case of this example we will be using a portable C150 logger connected to the PC via RS232. This logger has been set up as follows

```
SYNERNET
QUEUE
        LOGGER(S)
                ADD NEW LOGGERS
                        HOW MANY NEW LOGGERS TO ADD TO THE NETWORK: (1)
                        INPUT SITE NUMBER: (999)
                        INPUT LOGGER LETTER:(A)
                        INPUT LOGGER SERIAL NUMBER:(1613)
                        INPUT PARAMETER SET CODE:(A)
```

This results in the following logger being added:

```
                        999/A/1613/A
```

This logger can be reselected anytime by calling-up:

```
                        PICK FROM EXISTING LOGGERS
                                999/A/1613/A
```

Next, the logger needs to be configured so that it is recording the necessary information. This is accomplished with PARSET as follows:

```
SYNERNET
PARSET
        LOGGER(S)
                PICK FROM EXISTING LOGGERS
                        999/A/1613/A
```

EDIT

INTEGRATION PERIODS

NON-UNIFORM INTEGRATION PERIOD EDITOR

Table 3.1 is shows the integration periods for logger 999/A/1613/A. In general we see that one minute data are being sampled and captured to memory. When configuring a logger for the first time, it is best to check all integration periods with the non-uniform integration period editor. (NOTE: F10 is used to exit this system).

Next, the Watt meter channels are set up with

EDIT

WATT METER CHANNELS

The results of the session are shown in Table 3-2. In the case of this logger, one channel of 110 VAC single phase electricity is being monitored on CT0 using a 5 Amp CT. CT0 has been assigned to power register 0 and that volts, current, Watts, and volt-amps, are being recorded only for this channel. These are indicated by the "*" toggle that was activated with the F9 key for the STAtus (STA), (Hi), (Lo), Voltage, (V), Current (C), Power (P), and Apparent power (A). When one is done with this channel, exit by pressing F10 to the selection.

The analog channels are set up next by selecting

EDIT

ANALOG CHANNELS

The configuration is shown in Table 3-3. For logger 999/A/1613/A we can see that temperature is being recorded on analog channel 0, using a scale of 1, and an offset of 0 (this is the default for a 1000 Ohm RTD connected directly to the logger). This setting automatically produces output in Deg F as indicated.

Analog channel 1 is recording humidity using a 4-20 mA signal. In order to accomplish this we have previously placed a Synergistics 25A118-2 resistive header into the logger (for the portable logger we used a 200 Ohm precision resistor across the A1 to GND terminals.). This then allows the recorder to see DC volts using a scale of 31.25, and an offset of -25. The scale and offset allow for the 4-20 mA signal to read 0 to 100 % RH.

Next, the digital channels are configured as shown in Figure 3-13, and Table 3-4

EDIT

DIGITAL CHANNELS

In the case of this example, fluid flow is being recorded on digital channel 0 and thermal Btus are being recorded on digital channel 1. Units of one pulse equal to 1000 gallons, and one pulse equal to Btu x 10^6, respectively have been already assigned by the Btu transducer. They are, if effect, passed through the logger by choosing a state equal to 1.

Finally, the ordering of the configuration table is accomplished by using

EDIT
TSR MEASUREMENT NUMBERS

The configuration of logger 999/A/1613/A is now complete and is shown in Table 3-5 as follows:

COMPUTER ELEC.	KW 0	0
COMPUTER ELEC.	KV 0	1
COMPUTER ELEC.	VOLT 0	2
COMPUTER ELEC.	AMP 0	3
TEMPERATURE	AN 0	4
HUMIDITY	AN 1	5
FLOW-	DIG 0	6
BTU-	DIG 1	7

This is basically what is put to disk at each recording interval or Time Series Record (TSR). At this point we are ready to SAVE and/or PRINT the configuration tables. Tables 3-1 through 3-5 are the result of printing the configurations tables to a file.

The next step is to connect PARSET to the logger using

CONNECT
DIRECT SERIAL

PARSET will then respond with the appropriate message to tell us that Serial port 1 or 2 has been opened at the appropriate baud rate (this assumes that we have previously CONFIGured the PC on which SYNERNET is running in advance).

Actual communications with the logger is established with the TERMINAL command. At this point we have various different options. If we choose

TERMINAL
B READ REAL TIME DATA

We obtain the screen that is shown in Table 3-6 for logger 999/A/1613/A.

To download and/or look at TSR data we need to go back to the TERMINAL menu. This can be accomplished by pressing the ESC key. By choosing

TERMINAL
C READ TSR DATA

We can now look at and/or download Time Series Records (TSR) to the local PC for further processing. For example, after selecting this we see that we need to choose which TSRs to look at (or download), if we chose TSRs 900 through 925 we would have to entered:

```
ENTER STARTING TSR INDEX:  900    LAST TSR INDEX IS 1152
ENTER ENDING TSR INDEX:    925    PRESENTLY WORKING ON 976
(S)CREEN OR (F)ILE OUTPUT:  F
OUTPUT FILE NAME:     EXAMPLE.RAW
FILE TYPE: ASCII (R)EAL/(E)XPON, (W)K1 SPREADSHEET, (T)SR: R
HEADER TITLES? (N)ONE (A)SCII, (L)OTUS-IMPORT: N
```

This selects TSRs 900-925 to be recorded to disk on the PC in file EXAMPLE.RAW without headers and in real ASCII format. Table 3-7 is an example of what was recorded, from left to right the channels are

- Date
- Time
- TSR#
- Status
- KWH
- KVA
- VOLTS
- AMPS
- TEMP
- HUMIDITY
- FLOW
- BTU

3.3 SUMMARY.

This section of the workbook has introduced the use of a data logger. It is intended to be a first time survival guide. Additional details about the Synergistics C180 logger can be found in the documentation provided by the manufacturer. This section introduced the basics of connecting the logger to pulse, analog, and power signals, and provides an example session where a simple logger is configured to perform these tasks.

Survival commands for first time operation of the logger are provided, including the use of SYNERNET, QUEUE, CONFIG, and PARSET. An example configuration, setup, polling and retrieving of data session is also provided to walk the reader through his/her first session.

TABLE 3.1 *Configuration table for example logger showing integration periods.*

```
*********  Configuration for Logger: 1613  Parameter Set Code: A  *********

              ----- INTEGRATION PERIODS -----

  AM                  PM
From: 12  1  2  3  4  5  6  7  8  9 10 11 12  1  2  3  4  5  6  7  8  9 10 11
 To:   1  2  3  4  5  6  7  8  9 10 11 12  1  2  3  4  5  6  7  8  9 10 11 12

Flag:  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
Mins:  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

TABLE 3-2 *Configuration table for example logger showing Watt channels.*

```
               ----- WATT CHANNELS -----

Chan  Description      Search String  STA Hi Lo VMult  Amps  V C PR P A
---- ---------------- ---------------- --- -- -- ----- ------ - - -- - -
CT 0 COMPUTER ELEC.                    ON  A1 N1 1      5     * * 0  * *
CT 1                                  OFF  B1 N1 1    100     1
CT 2                                  OFF  C1 N1 1    100     2
CT 3                                  OFF  A1 N1 1    100     3
...
CT13                                  OFF  B1 N1 1    100    13
CT14                                  OFF  C1 N1 1    100    14
CT15                                  OFF  A1 N1 1    100    15

Chan                Field Notes
---- ------------------------------------------------------------
CT 0 electricity use of computers, etc.
CT 1
CT 2
CT 3
...
CT13
CT14
CT15
```

***TABLE 3-3** Configuration table for example logger showing analog channels.*

********* Configuration for Logger: 1613 Parameter Set Code: A *********

----- ANALOG CHANNELS -----

Chan	Description	Search String	STA	Scale	Offset	Units	T S G
A 0	TEMPERATURE		ON	1	0	Deg F	*
A 1	HUMIDITY		ON	31.25	-25	Volts DC	*
A 2			OFF	1	0		
A 3			OFF	1	0		
A 4			OFF	1	0		
A 5			OFF	1	0		
A 6			OFF	1	0		
A 7			OFF	1	0		
A 8			OFF	1	0		
A 9			OFF	1	0		
A10			OFF	1	0		
A11			OFF	1	0		
A12			OFF	1	0		
A13			OFF	1	0		
A14			OFF	1	0		
A15	NOT USED!		OFF	999	-999		

Chan	CType	Field Notes
A 0	1K RTD	Temperature
A 1	±5 VDC	Humidity
A 2	OFF	
A 3	OFF	
A 4	OFF	
A 5	OFF	
A 6	OFF	
A 7	OFF	
A 8	OFF	
A 9	OFF	
A10	OFF	
A11	OFF	
A12	OFF	
A13	OFF	
A14	OFF	
A15	OFF	

TABLE 3-4 *Configuration table for example logger showing digital channels.*

```
********* Configuration for Logger: 1613  Parameter Set Code: A *********

              ----- DIGITAL CHANNELS -----

Chan Description     Search String  STA  Scale     Units  TSR AVG RTS
---- ---------------- ---------------- --- ------- -------- --- --- ---
 D 0 FLOW-                 ON   1    galx1000   *
 D 1 BTU-                  ON   1    BtuxE6     *
 D 2                       OFF  0
 D 3                       OFF  0
...
 D13                       OFF  0
 D14                       OFF  0
 D15                       OFF  0

Chan              Field Notes
---- ------------------------------------------------------------
D 0
D 1
D 2
...
D13
D14
D15
```

TABLE 3-5 *Configuration table for example logger showing TSR channels.*

Description	Variable	Meas#
-----------	--------	-----
COMPUTER ELEC.	KW 0	0
COMPUTER ELEC.	KV 0	1
COMPUTER ELEC.	VOLT 0	2
COMPUTER ELEC.	AMP 0	3
TEMPERATURE	AN 0	4
HUMIDITY	AN 1	5
FLOW-	DIG 0	6
BTU-	DIG 1	7

***TABLE 3-6** Example of the TERMINAL, READ REAL TIME DATA screen.*

Logger(s)	Edit	Connect	Terminal	Other	Quit	
Chan	AMPS	VOLTS	POWER	PF	ANALOG	COUNT
0	3.0	120.0	0.250	0.70	'F 73.7	PC 300
1	-	-	-	1.00	V 2.258	PC 38
2	-	-	-	1.00	-	-
...etc...-	-	-	-	-	-	-
15	-	-	-	0.00	R 996.7	PC 0

***TABLE 3-7** EXAMPLE.RAW file recorded for sample session.* The columns shown are for Date, time, TSR#, Status, KWH, KVA, VOLTS, AMPS, TEMP, HUMIDITY, FLOW, and BTU

6/22/92	15:29: 0	900	"V "	0.224	0.388	120.640	3.218	72.566	45.845	0.000	0.000
6/22/92	15:30: 0	901	"V "	0.223	0.349	120.792	2.891	72.619	45.798	0.000	0.000
6/22/92	15:31: 0	902	"V "	0.210	0.340	120.868	2.815	72.619	45.798	0.000	0.000
6/22/92	15:32: 0	903	"V "	0.201	0.321	120.716	2.658	72.619	45.798	0.000	0.000
6/22/92	15:33: 0	904	"V "	0.202	0.336	120.868	2.786	72.672	45.798	0.000	0.000
6/22/92	15:34: 0	905	"V "	0.207	0.339	120.792	2.810	72.724	45.751	0.000	0.000
6/22/92	15:35: 0	906	"V "	0.198	0.321	120.716	2.658	72.672	45.704	0.000	0.000
6/22/92	15:36: 0	907	"V "	0.298	0.440	120.184	3.669	72.724	45.798	0.000	0.000
6/22/92	15:37: 0	908	"V "	0.214	0.372	120.868	3.086	72.672	45.798	0.000	0.000
6/22/92	15:38: 0	909	"V "	0.199	0.322	120.868	2.667	72.619	45.798	0.000	0.000
6/22/92	15:39: 0	910	"V "	0.210	0.336	120.792	2.778	72.672	45.845	0.000	0.000
6/22/92	15:40: 0	911	"V "	0.229	0.362	120.716	2.998	72.672	45.892	0.000	0.000
6/22/92	15:41: 0	912	"V "	0.225	0.361	120.716	2.990	72.672	46.033	0.000	0.000
6/22/92	15:42: 0	913	"V "	0.257	0.390	120.564	3.237	72.882	46.127	0.000	0.000
6/22/92	15:43: 0	914	"V "	0.253	0.384	120.716	3.185	72.882	46.221	0.000	0.000
6/22/92	15:44: 0	915	"V "	0.260	0.397	120.564	3.296	72.830	46.174	0.000	0.000
6/22/92	15:45: 0	916	"V "	0.253	0.381	120.488	3.166	72.882	46.174	0.000	0.000
6/22/92	15:46: 0	917	"V "	0.268	0.404	120.488	3.353	72.988	46.174	0.000	0.000
6/22/92	15:47: 0	918	"V "	0.252	0.381	120.564	3.158	72.830	46.221	0.000	0.000
6/22/92	15:48: 0	919	"V "	0.268	0.409	120.488	3.393	72.988	46.268	0.000	0.000
6/22/92	15:49: 0	920	"V "	0.247	0.387	120.716	3.207	72.935	46.268	0.000	0.000
6/22/92	15:50: 0	921	"V "	0.259	0.393	120.716	3.255	72.988	46.315	0.000	0.000
6/22/92	15:51: 0	922	"V "	0.252	0.387	120.640	3.210	72.988	46.315	0.000	0.000
6/22/92	15:52: 0	923	"V "	0.259	0.395	120.564	3.279	72.935	46.362	0.000	0.000
6/22/92	15:53: 0	924	"V "	0.253	0.398	120.716	3.297	73.093	46.408	0.000	0.000
6/22/92	15:54: 0	925	"V "	0.244	0.383	120.640	3.178	73.093	46.455	0.000	0.000

FIGURE 3-1 *Typical electrical connection of the logger* (Reproduced with permission: Synergistics 1990).

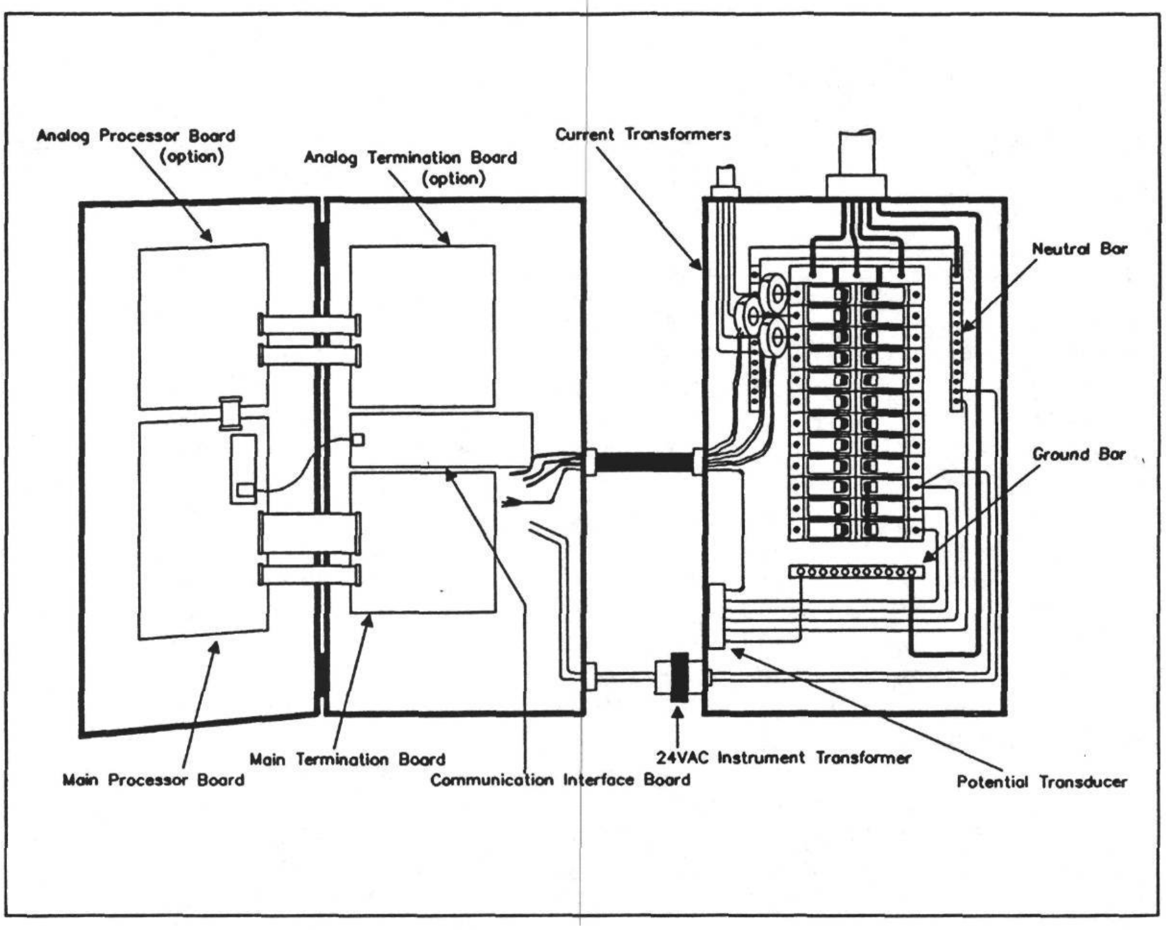


FIGURE 3-2 Power and digital termination board for the logger (Reproduced with permission: Synergistics 1990).

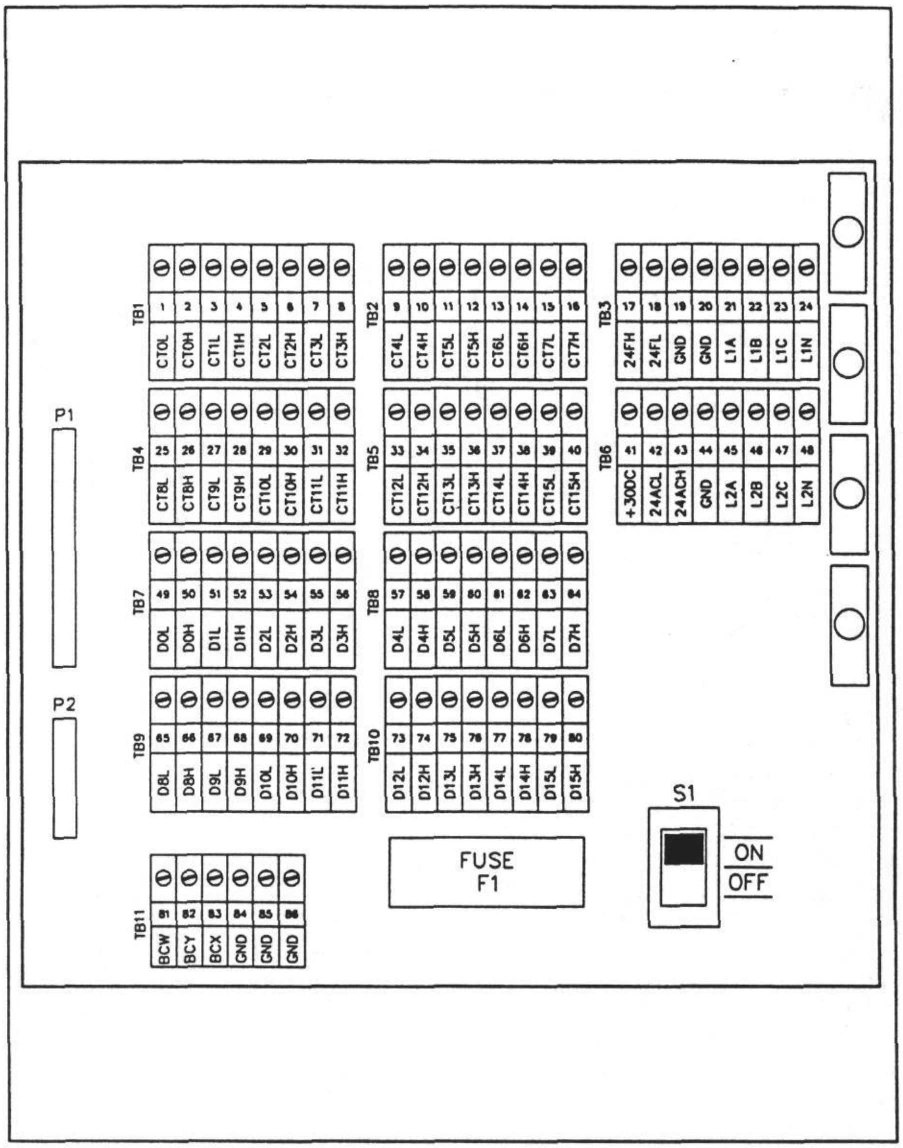


***FIGURE 3-3** Analog termination board to the logger* (Reproduced with permission: Synergistics 1990).

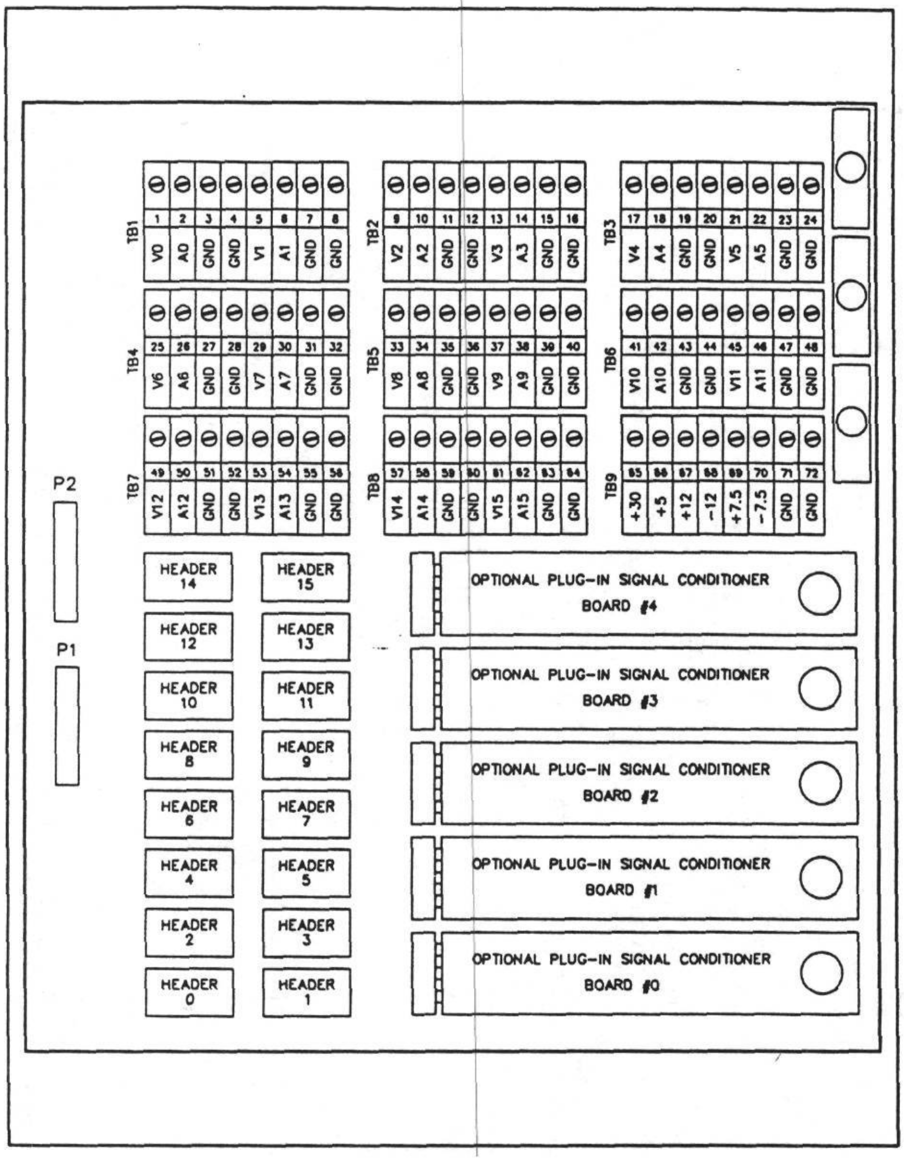


***FIGURE 3-4** Connecting a digital pulse circuit to the logger* (Reproduced with permission: Synergistics 1990).

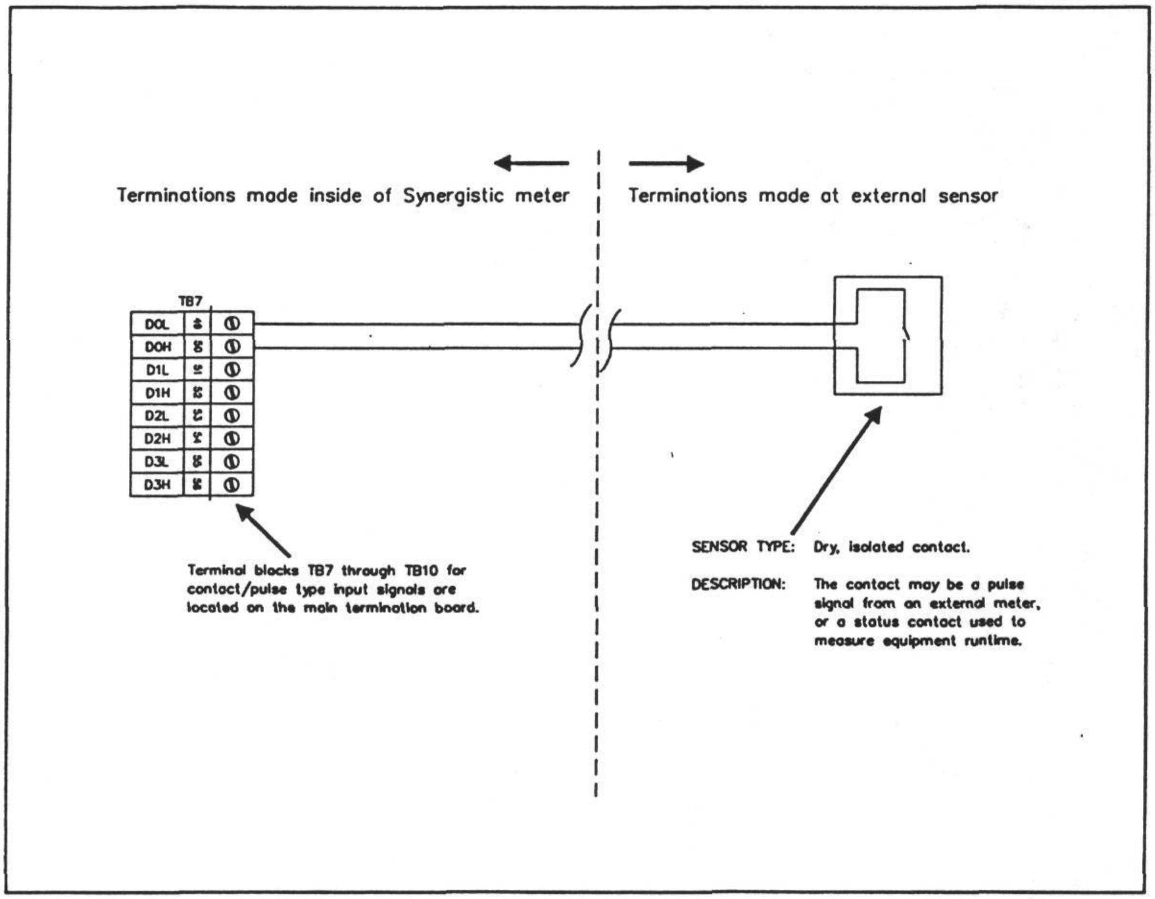


***FIGURE 3-5** Connecting a 2-wire RTD sensor load to the logger* (Reproduced with permission: Synergistics 1990).

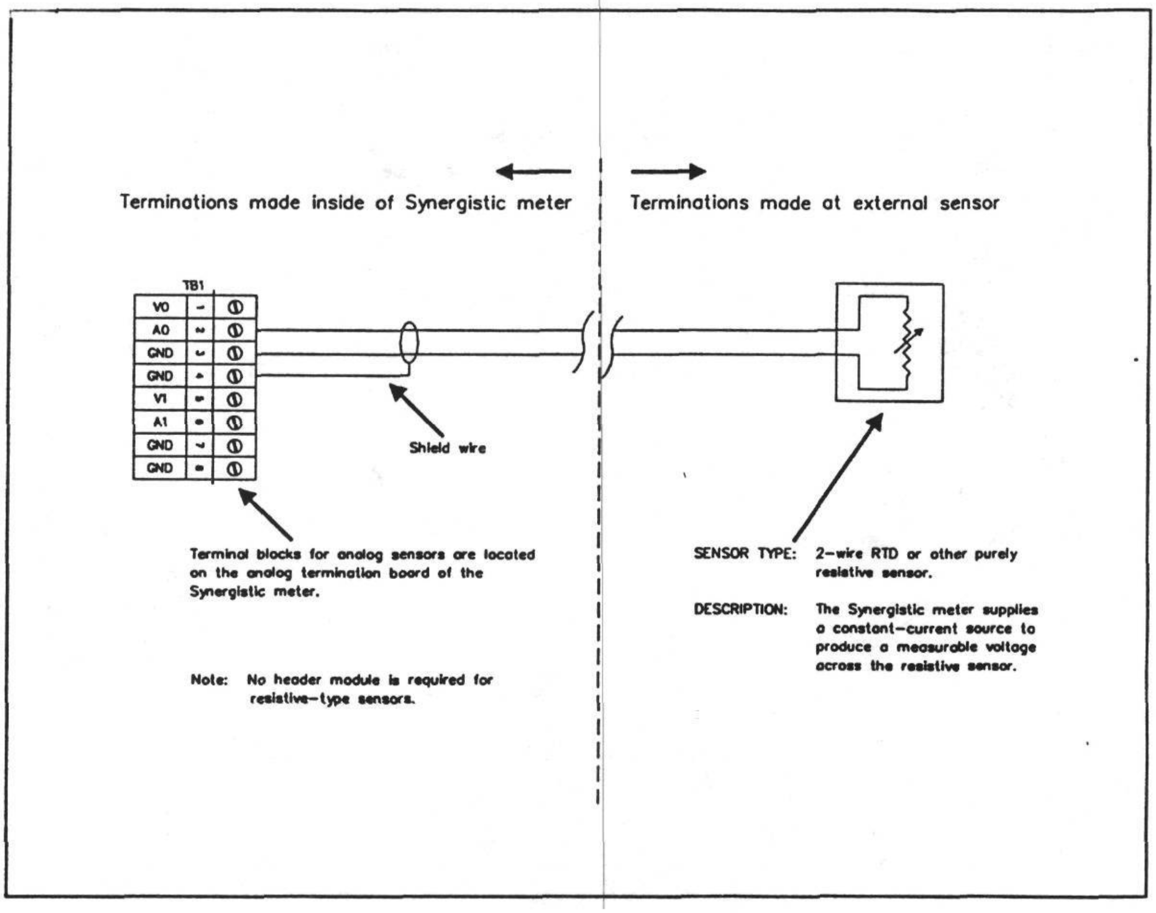


FIGURE 3-6 *Connecting a 4-20 mA signal from a sensor transducer* (Reproduced with permission: Synergistics 1990).

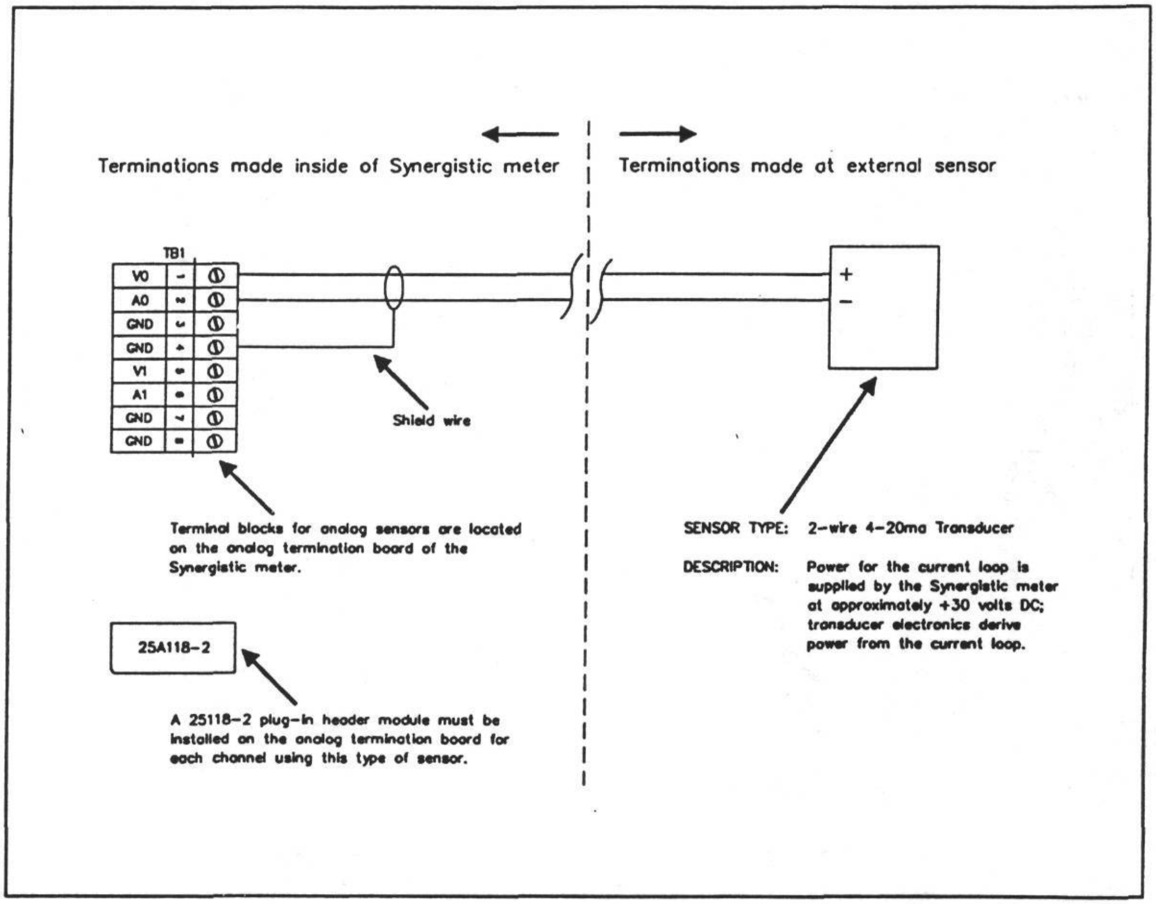


***FIGURE 3-7** Connecting multiple current transducers (CTs) to the logger using a summing module* (Reproduced with permission: Synergistics 1990).

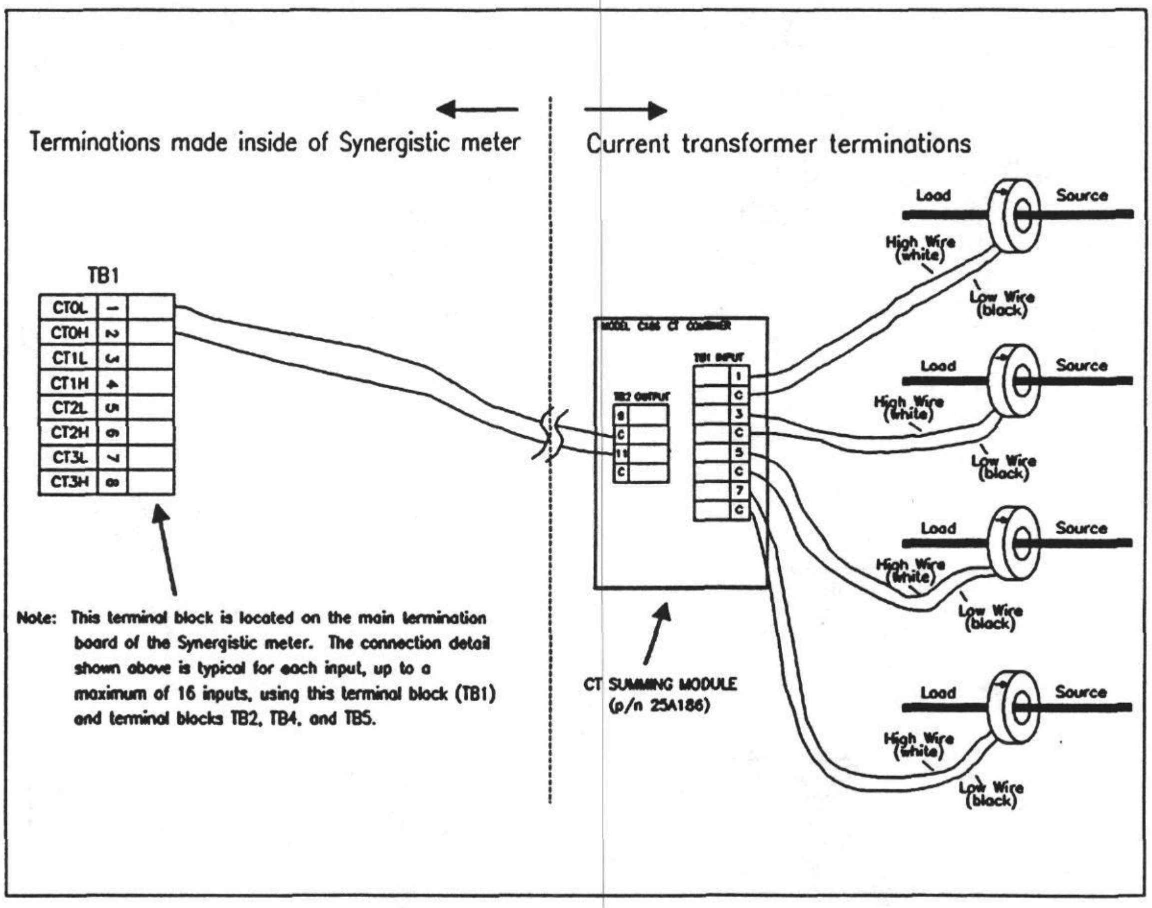


***FIGURE 3-8** Connecting the potential transducer (PT) from a 3-phase 4-wire circuit to the logger* (Reproduced with permission: Synergistics 1990).

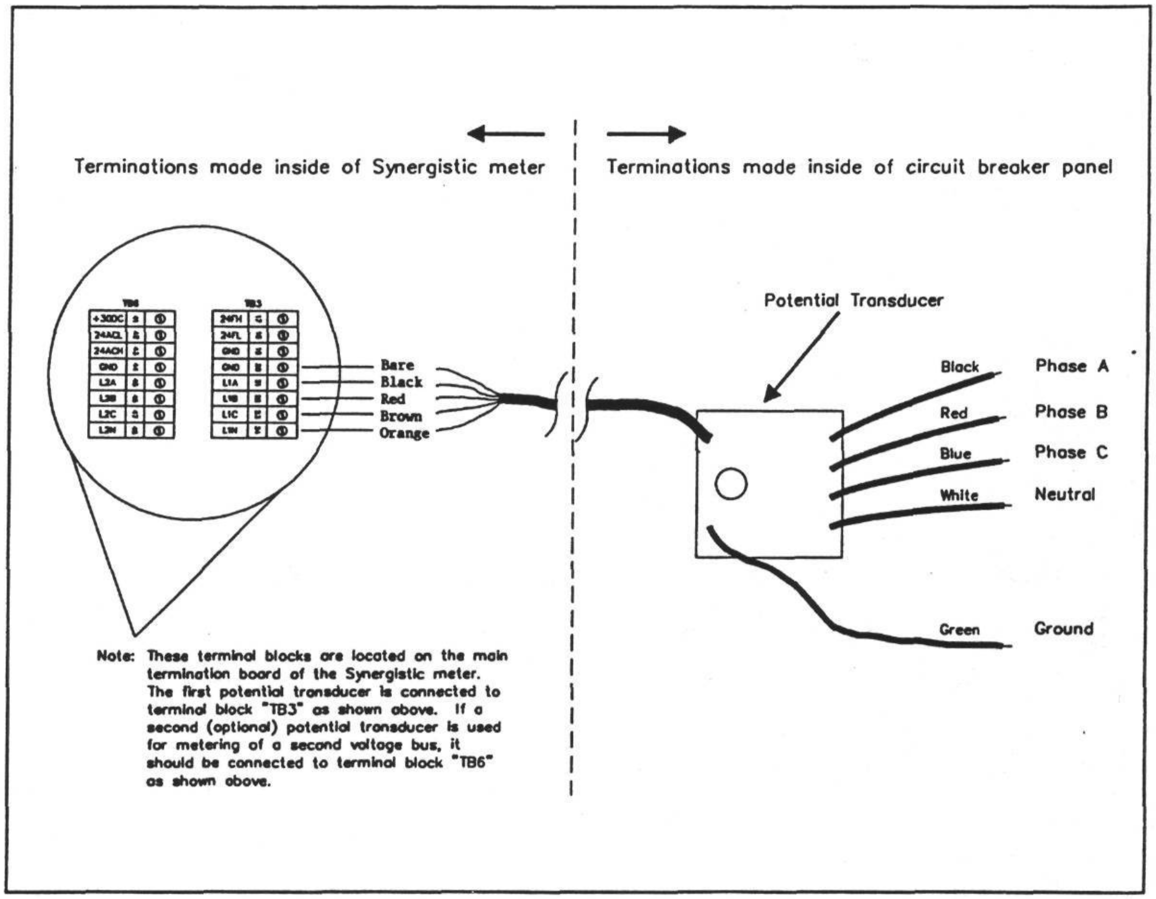


FIGURE 3-9 *Main menu diagram for the SYNERNET program.*

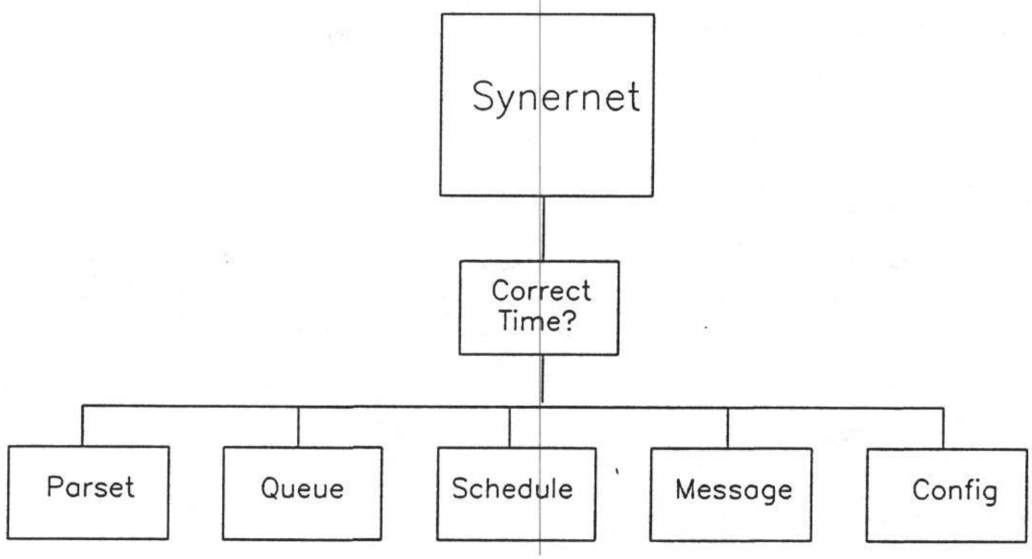


FIGURE 3-10 *CONFIG(uration) settings for typical software setup.*

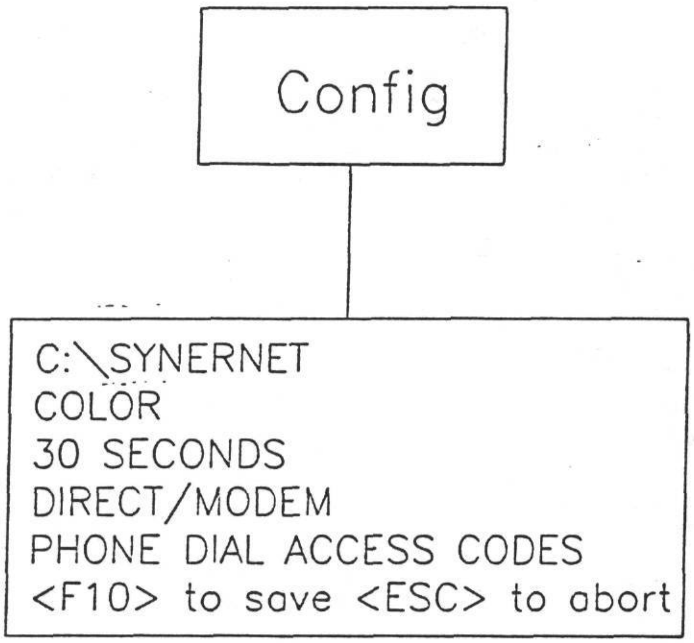


FIGURE 3-11 Diagram for QUEUE program.

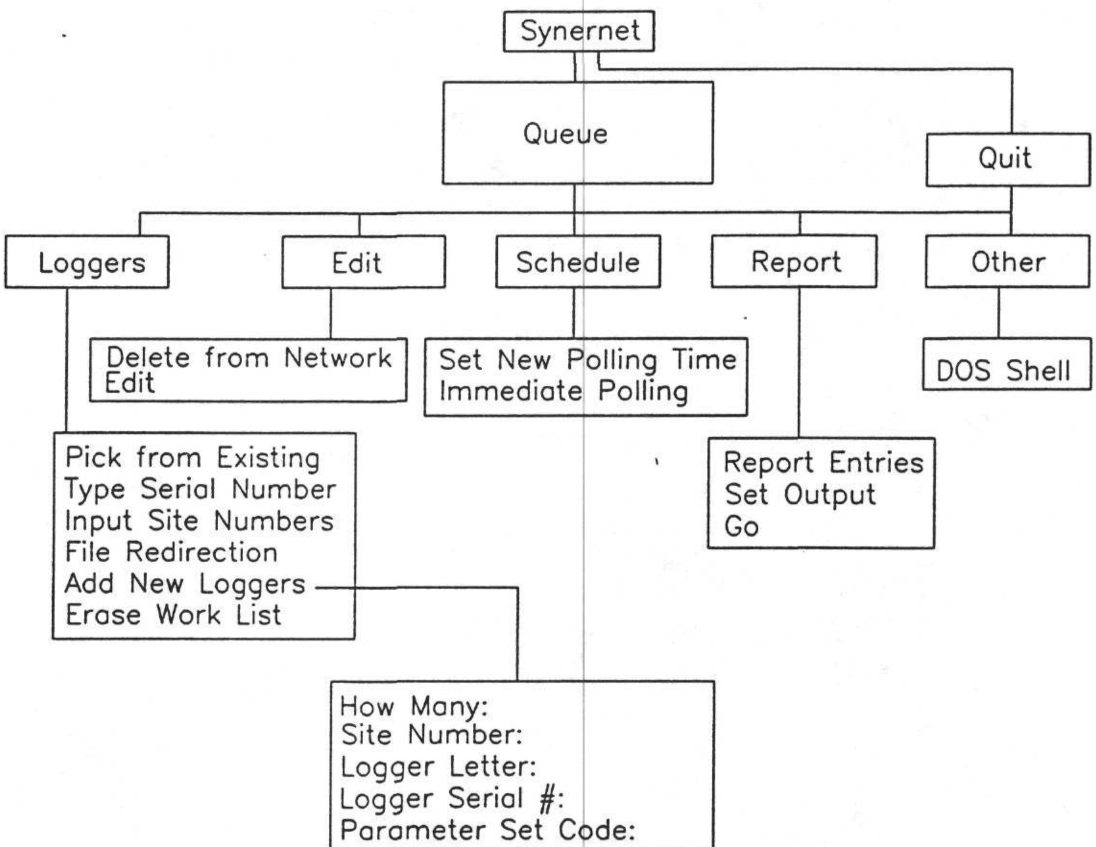


FIGURE 3-12 Diagram for the PARSET program.

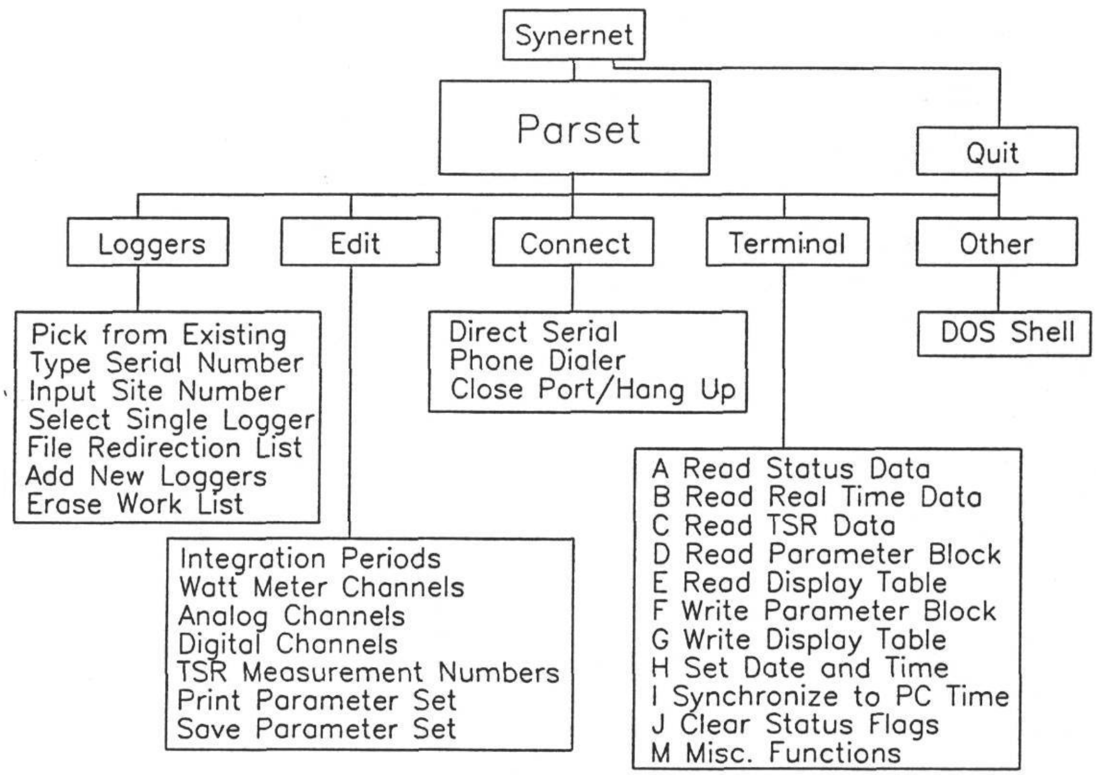
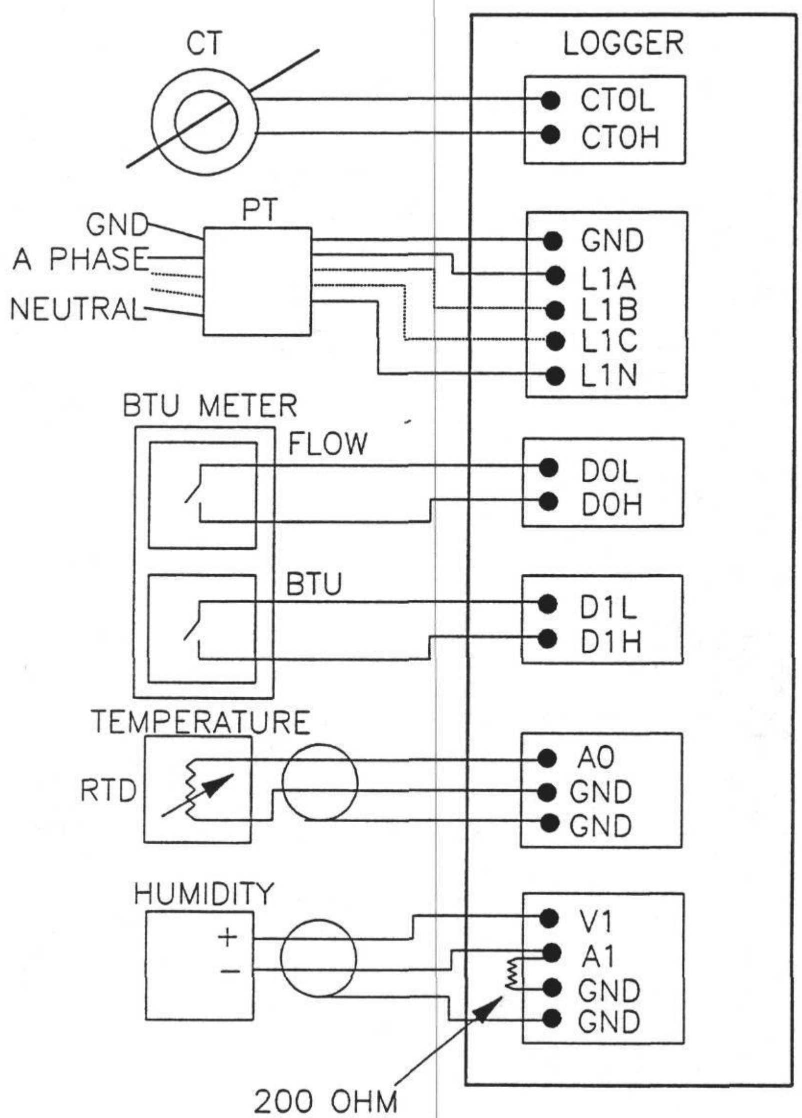


***FIGURE 3-13** Diagram of an example logger set-up.*



4.0 WHAT TO DO WITH THE DATA

This section provides the reader with some helpful hints about what to do with the data once it has been collected from a logger. This section describes a collection of routines used to process and plot data collected from Synergistics loggers used by the Texas LoanSTAR Program over the last three years on a weekly basis. Instructions and sample code are provided on disk for developing inspection and summary plots, and 3-D plots using a combination of public domain data processing toolkits and inexpensive commercially available plotting software. Table 4-12 contains a listing of the routines that are provided on the disk.

4.1 PROCESSING/PLOTTING SYNERGISTICS DATA

This section describes a collection of routines that are used to process and plot data collected from data loggers. These routines have been used by the Texas LoanSTAR Program over the last three years on a weekly basis to create a set of *inspection plots* which can be used as a primary quality control measure. One of the major goals of LoanSTAR has been to provide a fully functional system for presenting measured building energy data at a minimum of both cost and effort. To these ends, the system utilizes a collection of inexpensive commercially available products and public domain packages, as well as a set of public domain software filters written in-house to knit the streams together.

Controlling batch files are used to call the routines in sequence; once a production mode is established for creating the plots for a particular building, only a few keystrokes are required to actually create the graphic report. As each building being measured possesses some unique properties, an inherent amount of tailoring of these routines is required. Most of the effort required to run and maintain this system is in the initial setup.

The enclosed diskette contains a sample set of data for a typical university building, an associated weather file, and the routines used to process the data. This processing includes:

1) Automated quality control checks of all data channels against static lower and upper bounds.

2) Insertion of missing records with bad data markers (-99).

3) Creation of time series graphs of all channels, grouped twelve graphs per page.

4) Creation of summary plots.

The processing stream makes these assumptions:

1) Data are being collected on an hourly basis.

2) Each data file to be processed contains *exactly one week* worth of hourly records.

3) Each site (data logger) has an associated three-digit code. The example used herein is site 101 - The University Teaching Center at the University of Texas in Austin.

4) The raw data recorded from the Synergistics logger has been stored into a file using real numbers and no headers from the Synergistics software (10192168.RAW).

5) The file name used to record the raw data follows the strict format of XXXYYDDD.RAW where XXX is the three-digit site code, YY is the year, and DDD is the number of the day during which the data were collected. Collectively, YYDDD is known as the *Julian date*. As an example, the raw data file included on the distribution diskette is 10192168.RAW (Table 4-1). This is data for site 101. Because it was collected on 92168 (the 168th day of 1992 or June 15, 1992), this file contains data for the period beginning 92161 (June 8, 1992 at midnight) and ending 92167 (June 15, 1992 at 11:00 p.m.).

6) To print summary plots, a weekly weather file containing hourly data for the region is present (20292168.WEA).

7) The commercially available graphing program GRAPHER (Golden 1990) is used to create the plots.

8) The public domain programs ARCHIVE and COLS (Feuermann and Kempton 1987), and GAWK(FSF 1989) (included on the distribution diskette) are used for quality control and data manipulation.

9) The subdirectory \TEMP has been created prior to running the routines.

The following sections discuss the methodology of these routines as well as possible modifications for plotting other metered data.

<u>Installing the sample routines</u>

To run the sample data file through the processing stream, simply copy all files from the distribution diskette into the same directory on the hard disk.

NOTE:

- *The directory containing GRAPHER must be placed on the path in the AUTOEXEC.BAT file, and GRAPHER should be configured to run out of the current directory and plot in Postscript output.*
- *GRAPHER should also be configured to run in landscape mode (plot rotation = yes).*
- *A \TEMP subdirectory must be created on the hard disk if it does not already exist.*

- *A copy of the ARCHIVE manual by Feuermann and Kempton is included in the appendix to this workbook.*

Preparing data from time series channels from raw Synergistics data with R2A.BAT.

Given this set of filters and programs, a rudimentary quality control range check can be performed and a full set of time series plots can be created with a simple command line operation. To perform the quality control and produce plots, type:

R2A.BAT 101 92168 90001 <CR>

This command calls the controlling batch file R2A.BAT to begin the process as shown in Figure 4.3. The parameters passed to R2A.BAT include the three-digit logger code (101), a Julian polling date (92168), and a channel table descriptor (90001). R2A.BAT uses the logger code and julian date to understand which file to process.

R2A utilizes an ARCHIVE channel table: which is a data dictionary that attaches static high/low bounds, English language descriptions and scaling factors to each data column. In a long term monitoring project, this channel table might change significantly during the course of the project; therefore, different channel tables can be written or changes added to the same channel table. The channel table descriptor tells R2A.BAT which channel table is current for the data being processed.

The output of this scheme includes the flat file 10192168.ACS, which is incorporated into the LoanSTAR database, and two pages of Postscript output, 101ONE.OUT and 101TWO.OUT These pages contain time series plots of all the channels being monitored at site 101. Examples of these pages are given as Figures 4-2a and 4-2b.

A flowchart for R2A.BAT is given as Figure 4-3. A copy of R2A.BAT is included on the distribution disk.

Briefly, these steps are performed by R2A.BAT:

1) The GAWK script RAW2DAT.AWK is called to preprocess the raw data 10192168.RAW for quality control checks. The quality control is performed by the public domain program ARCHIVE (Feuerman & Kempton, 1989), which is unable to understand some of the characters that the Synergistics software leaves in the 10192168.RAW file. An example of Synergistics data is given as Table 4-1. The output of RAW2DAT is given as Table 4-2. Notice date/time columns have been adjusted, and certain characters have been stripped-out of the file (e.g., "/", ":", etc.).

2) The output of RAW2DAT and the site's ARCHIVE channel table are fed into ARCHIVE for static high/low range checking. The ARCHIVE channel table 10190001.CHT for site 101 is given as Table 4-3. Example output from ARCHIVE is given as Table 4-7. ARCHIVE will report any offending data readings in a logfile and will replace such readings in the data with a

"bad data" marker (Table 4-4). Currently, this marker is -99. ARCHIVE automatically appends the DOS file extension .ACH to the filename. For the example dataset provided, this step will have created the file 10192168.ACH.

An ARCHIVE channel table is created for each site and contains the instructions that ARCHIVE uses to process the data from each site. In Table 4-3, the ARCHIVE channel table 10190001.CHT is shown that process the data from site 101. The first four lines of the channel table are labels for the columns below. The "--------" column alignment guide indicates to the user how many characters to include in each parameter, and uses a blank as a separating delimiter. The line beginning with "#" contains special characters that tell ARCHIVE what kind of data it is processing, and what to use as a missing variable (the default is -99).

- The first eight characters are the date that the parameters are to be applied. Excluding the last line, this is "04/23/92" for site 101 which is the most recent date for this parameter set.

- The next variable is the time, in this case "00:00". This is instructing ARCHIVE to begin processing on April 23, 1992 at midnight.

- Next are the line number and column number of the input channel. These are followed by the ARCHIVE output column number. A "0" value is essentially a comment line and does not appear in the .ACH file.

- Following the ARCHIVE column position indicator is an eight character descriptor of the channel. This is followed by another twelve character descriptor of the ARCHIVE units and a six character codeword for the ARCHIVE output format.

- The next two variables contain the conversion codeword and conversion constants. The conversion codeword is an integer from 1 to 31 and instructs ARCHIVE whether or not to perform conversions on the incoming data. Conversion code "0" will place a missing variable into this column, code "1" is an identity code that allows the value to pass through ARCHIVE untouched, code "2" is a linear transformation that requires two constants (i.e., slope and intercept), and so forth.

- The last three columns contain the error code, error constants, and channel description. The error checking code is an 1=on, 0=off code that initiates the high/low limit checking which makes use of the high/low limit values that immediately follow.

In the 10190001.CHT channel table in Table 4-3, there are 24 lines of input. The first line

    04/23/92 00:00 1 0 0 Begin UTC   Beginning Date

is basically a comment line that does not appear in the output. The next line

04/23/92 00:00 1 1 1 Bldg.# XX I3 2 0 101 0 0 *Good starting 92.19*

places the site number "101"in the first column of the ARCHIVE output. This is done by using a linear transformation of slope = 0 and intercept = 101.

The next six lines

04/23/92	2
⇓	⇓
04/23/92	6

create the second through seventh columns in the output file. The second, third, and fourth columns in 10192168.ACH are the month, day and year that are simply passed through ARCHIVE without change.

- The fifth column is the Julian date (92161), that is calculated by ARCHIVE using the first, second, and third input columns.

- The sixth column is the decimal date (4543.000) that is calculated by ARCHIVE. The decimal date is a combined date and time stamp that is an offset number of days and hours from January 1, 1980. It is similar to the @DATE(YR,MO,DAY) function that is used in many spreadsheets.

- The seventh column is the hour of the day using military notation (i.e., 0 to 23 hours).

- Columns eight through eighteen in 10192168.ACH all contain monitored data from the UTC building.

As can be seen in Table 4-7, many additional columns have been appended to the data. These columns are:

Site number. Because these files are stored as ASCII text, it is very easy for records from other sites to be mistakenly inserted into a dataset. This simple site number at the beginning of each line is a first step to ensure that the data recorded in this file really belong here.

Month, Day, Year. This is recorded in the file so that people can understand the dates.

Julian date. This has been discussed previously.

Decimal date. Although this looks bizarre at first, the decimal date is a very handy way for a computer to understand time and graph data points as a time series. The integer part of the decimal date is merely an offset number of days from January 1, 1980. The decimal part is the hour of the reading divided by 24. This date format guarantees different timestamps for all data points including leap years. A time series graph based on hourly indices is difficult to handle and plot because of the cyclic nature of the clock (20, 21, 22, 23, 0, 1, 2, 3, etc.).

Time. This is recorded in military units to differentiate between a.m. and p.m.

3) The .ACH file is fed to the program MISSING. This program scans the timestamps and insert records and appropriate bad data markers for any missing records. When a logger loses power in the field, it stops recording TSRs, and begins recording TSRs when the power is restored. However, a hole will exist in the data for those periods when the power was off. This hole is filled to aid in file merging and in graph readability. The output of MISSING uses the file extension .ACS. This is the ASCII flat file from which most of the work at LoanSTAR gets done. When there are no missing data there is no difference between an .ACH and .ACS file except the extension.

4) The final task of R2A is to call the graphing routines found in the batch file 101GRAPH.BAT. This program is sufficiently complicated and merits discussion on its own.

Using GRAPHER to Create an Individual Graph.

GRAPHER is one of many commercially available general purpose graphics software. GRAPHER is very useful for rapidly plotting data because of its flexibility, overlay and programmable batch mode operation. GRAPHER is actually composed of several subprograms as shown in Figure 4-1a. The most important of these (once configured) are the VIEW and PLOT programs. VIEW allows one to quickly preview a graph that has been created. PLOT translates GRAPHER 's .PLT file into device-specific plot instructions.

In general, to produce a plot with GRAPHER , one needs data (.DAT) and plotting instructions (.GRF). GRAPHER also allows for additional customization with axis (.AXS), grid (.GRD), dividing line (.DIV) and text (.TXT) files.

Figure 4-1b shows the result of processing the T1017.GRF GRAPHER instruction file. Table 4-5 contains a summary of the graphic instructions contained in the T1017.GRF file. Table 4-6 is the T1017.GRF file that GRAPHER produces. From Table 4-5, one can see that input file T101.DAT is being used and that a linear X-Y plot is being produced using the sixth column (F) for X and the fourteenth column (N) for the Y variable. GRAPHER produces a time series graph since the X variable is actually the decimal date and a solid line without symbols is being used to plot the data. Each graph that is to be plotted requires a .GRF file. The use of GRAPHER to produce weekly inspection plots is reasonably efficient because the same .GRF file (modified slightly) can be used with each week's data.

Creating graphs using 101GRAPH.BAT.

101GRAPH.BAT is another controlling batch file. Its function is to automatically produce a set of time series plots (Figure 4-2a and 4-2b), one per channel being reported by the logger. A flowchart for 101GRAPH.BAT is given as Figure 4-4. The copy of the actual code can be obtained by printing the 101GRAPH.BAT file that is included on the distribution disk.

Briefly, these steps are performed by 101GRAPHBAT:

1) Copy a temporary version of the 10192168.ACS file into the \TEMP directory and rename the file to T101.DAT. This is required because GRAPHER only takes files with the .DAT extension as input.

2) Call the GAWK script 101DATE.AWK to determine the beginning dates in the dataset. This script automatically writes the batch file 101CHGRF.BAT.

3) Call 101CHGRF.BAT. This uses the GAWK script 101CHGRF.AWK and the dates found in 101DATE.AWK to change the .GRF files for each plot. These files need to be changed to start the time line (the X axis) at the correct spot for each week. As each GRAPHER file is modified, it is written into \TEMP.

4) For each .GRF file in \TEMP, call GRAPHER. The output is a device independent .PLT file.

5) Format each page. To print twelve graphs per page, the .PLT files need to be shrunk and pasted together. This is accomplished with a simple set of scale/translate files and the DOS copy command. The scale/translate files work as follows:

- A.PLT: Shrink. Move to the lower left corner (only one shrink command is needed).
- B.PLT: Move up one row.
- C.PLT: Move to the right one column and down two rows (back to the bottom of the page).

Therefore, a full page of twelve plots (i.e., #1, #2, ...#12) is created by appending all of these together:

A.PLT + #1 + B.PLT + #2 + B.PLT + #3 + C.PLT + #4 + B.PLT + #5 + B.PLT + #6 + C.PLT + #7 + B.PLT + #8 + B.PLT + #9 + C.PLT + #10 + B.PLT + #11 + B.PLT + #12.

6) For each page of twelve graphs, use the GRAPHER PLOT program to create a Postscript .OUT file.

7) Clean out all the temporary files.

Modification of routines

In order to modify the example routines to produce plots of channels from another logger, the following steps must be taken:

1) Assign a three digit code to the logger. For the purposes of these instructions, we'll use the generic designation XXX.

2) Using the Synergistic PARSET program, make a printout of the logger's parameter set for this site. This is accomplished with the EDIT PRINT command after a logger has been selected. The last page produced by this report is a list of the channels that the PARSET software is using, as well as the left-to-right ordering of the columns in which they will appear in each TSR.

3) Using the last page of the parameter set report, modify the example ARCHIVE channel table (Table 4-3) to reflect the new site. This can be accomplished with any ASCII editor. These modifications include:

- The site number (line 2 after the header).
- The start date (first column of all lines after the header).
- The name of each channel.
- The error constants for each channel.
- The description of each channel.
- Scaling of any channel that has not been previously scaled by PARSET.

Save this file as XXX90001.CHT.

4) Modify 101DATE.AWK to utilize site XXX instead of 101. Save this file as XXXDATE.AWK.

5) Modify 101CHGRF.AWK to utilize site XXX instead of 101. Save this file as XXXCHGRF.AWK.

6) 101CHGRF.BAT does not need to be changed. This routine will be written automatically at runtime.

7) For each channel in the parameter set, a template GRAPHER .GRF file needs to be written. The easiest way to do this is to modify the T1011.GRF through T10116.GRF files. This can be accomplished by either using GRAPHER interactively or by hand-editing the .GRF files. The following items must be changed:

- Data file name from T101 to TXXX (in line 3).
- The site/date designation from "Site 101 Beginning" to "Site XXX Beginning" (line 16).
- The actual description of the channel being plotted (line 24).

8) Modify and rename 101GRAPH.BAT to utilize site XXX. This includes changing the user diagnostics that appear on the screen both at the beginning of the script and upon termination, but it mainly consists of tailoring the middle section to knit together the page sections that print all the graphs cleanly. As has been stated previously, the given scaling and translation factors in A.PLT, B.PLT, and C.PLT can yield up to twelve graphs per page. Additional pages that might be required can be pasted into the routines by copying the lines which create page XXXONE.PLT and using a different destination filename.

4.2 CREATION OF SUMMARY PAGES FROM RAW SYNERGISTICS DATA AND AREA WEATHER DATA

Because each building usually has a unique parameter set summary plot pages have been created to produce a generalized scheme for quickly inspecting data collected from multiple buildings. A summary plot page contains whole building information presented in a standard orientation.

The motivation for creating such a page is twofold: first, in many buildings whole building readings are often recorded on multiple channels (e.g. A, B, and C phases); second, unless a very rigid channel selection is followed each time, it is very difficult to get the same graph appearing in the same location on the standard time series pages over several buildings. Summary plot pages decrease the time required during plot inspection because they present whole-building data. It was found early on in the LoanSTAR Program that pages such as these are tremendously helpful for visual quality control.

An example summary page is shown in Figure 4-5. The first row of the summary page always contains a time series plot of whole building electric for the site, as well as weather time series data (outdoor dry bulb temperature, relative humidity, and solar radiation) for the region. In the LoanSTAR program, one or more loggers may share the same weather station. In the case where a logger does not have its own weather station, weather data from a nearby site must be merged in from an outside file.

The second row contains time series graphs of building chilled water consumption, hot water consumption or steam condensate, submetered electrical points, and lighting where applicable. The third and final row contains scatter plots of the same data points in the second row plotted against outdoor dry bulb temperature for the region.

To create a summary plot page for the example dataset provided, type:

UTSUMM 101 92168 <CR>

This invokes the controlling batch file UTSUMM.BAT, and tells it to print a summary of site 101, using the julian date of 92168 as its target. The output is the file 101SUMM.OUT which contains the summary page in Postscript form. The flowchart of UTSUMM.BAT is given as Figure 4-6.

The methodology behind the creation of summary pages is almost identical to that of the standard time series plots, preceded by two steps:

1) A step to create summary pseudo-channels. This is needed in buildings with multiple electric or chilled water feeds to create a temporary channel which totalizes all the subfeeds. This is accomplished for site 101 with a call to the GAWK script 101SUMM.AWK.

For example, included with this workbook 10192168.ACS is passed to 101SUMM.AWK and produces TS101.DAT in the current directory. Within 101SUMM.AWK Motor Control Center (MCC) channels #8, 9, 10, 11, 12, 13, 19, 20, 21, and 22 are added together to form a total MCC channel, excluding any values that are missing (-99). The TS101.DAT output file then contains values for the Motor Control Center (MCC), whole-building electricity (ELEC), whole-building steam use (HW), and whole-building chilled-water use (CW). The whole-building steam is multiplied by 9.075 to convert from gallons to Btus.

2) A step to merge in regional weather data for the same week. Assuming both the building data and the weather data have been filtered through the MISSING program, this step can be accomplished with a simple call to COLS. COLS is one of the helpful toolkits that comes with ARCHIVE.

The output file TS101.DAT is then merged together with local weather data with the command:

```
COLS ^TS%1.DAT ^202%2.WEA A1:5 B8:11 > \TEMP\TS%1.DAT <CR>
```

This calls COLS with the input files TS101.DAT and 20292168.DAT. Columns 1 through 5 from TS101.DAT and 8 through 11 from 20292168.DAT are merged together to form TS101.DAT in the subdirectory \TEMP, which is immediately followed by a deletion of TS101.DAT in the current directory because it is no longer needed.

The remaining steps that are performed in UTSUMM.BAT are as follows:

3) Call the GAWK script 101DATE.AWK to determine the beginning dates in the dataset. In a similar fashion to 101GRAPH.BAT, this creates the batch file 101CHGRF.BAT.

4) Call 101CHRGF.BAT, which uses 101CHGRF.AWK to change the TS*.GRF files for each plot. Modified graphs are then written to \TEMP.

5) Change directory to \TEMP. Then for each .GRF file in \TEMP, call GRAPHER.

6) Format the summary page using a combination of shrink and paste commands as shown.

7) Plot the Postscript file and clean-up.

<u>Modification of summary plot routines</u>

Prior to modification of summary plot routines, it is assumed that the process for creating the standard time series plot pages is in order and functional. To modify the example routines to produce summary plots of channels from an arbitrary logger, the following steps should be taken:

1) Modify each template GRAPHER file, TS1011.GRF through TS10112.GRF, to reflect site XXX. This is identical to changing the original files when creating the standard pages.

2) Modify UTSUMM.BAT to do the following:

a) Report the correct site number in the user diagnostics upon entrance and exit.
b) Merge the weather data from the correct region/file.

3) Create a GAWK summary script similar to 101SUMM.AWK. The contents of this script will be completely site specific.

4.3 CREATING A 3-D GRAPH USING LOTUS 123 AND INTEX SOLUTIONS 3D GRAPH.

3-D graphs have been shown to be useful in displaying schedule-related whole-building and end-use energy profiles. However, it is not always easy to create useful 3-D plots on a PC because certain software packages require that data be placed in a special format prior to processing. One such combination of processing schemes is shown in Figure 4-7. Columnar data are plotted with the Intex Solutions 3-D plot package that can be attached to Lotus 123 on a PC.

To facilitate the creation of 3-D plots a special routine was created to convert COLumnar data into ROW format to produce a 3D plot -- COLROW3D (1991). With this routine two columns of ASCII data are fed to COLROW3D where they are reformatted into a row-wise matrix to allow for importing into 123 for plotting with the 3-D graphics add-on package. To facilitate this easily in a batch mode previously compiled 3-D plot instructions can be used in a 123 macro file as shown in Figure 4-7. Output from 123 consists of .PIC files that can be plotted or passed on to additional programs for further processing. This next section describes how to use the software to produce 3-D surface plots with the Lotus 123 add-on package that is available from Intex Solutions. The reader is referred to the Lotus 123 manual or the Intex Solutions 3-D graphics manual for further information about plotting the 3-D graphs.

<u>Using the COLROW3D Columnar to Row Data Processing Routine.</u>

COLROW3D is a columnar data manipulation program which processes hourly energy consumption data to produce a "new" file containing a spread sheet compatible data matrix. COLROW3D compresses each day's worth of data into one row in the matrix. For example, a leap year's worth of hourly data (8784 lines) will be compressed down to just 367 lines!

The output file generated by COLROW3D can be used in conjunction with Lotus 123 and Intex Solution's 3D-Graphics add-on package to produce a three dimensional (3D) picture of energy consumption versus day of year and time of day. COLROW3D also creates a .LOG file containing information about the run and any erroneous data found. The COLROW3D program is written in ANSI Standard C. The source code is provided on the diskette that accompanies this workbook.

Keywords:

Source:	COLROW3D's input filename.
Dest or Destination:	COLROW3D's output filename.
LOG:	COLROW3D's log file.
Method or Option:	Code used to process input data.
3D extension:	Output filename extension.
3D Graph:	Three dimensional graph.
Decimal date:	Data string used to express date and time of data together.
kWh/h:	Hourly energy consumption data.

Requirements:

For COLROW3D*:* IBM PC compatible computer with at least 128K memory
MS-DOS 2.11 or greater

Input file(s):

The original energy consumption file contains two columns of data: Date (day of year and time of day expressed as a single decimal date string), and consumption (expressed in units between -999.9 and +9999.9). The data should be separated by a space from the decimal date and can be of real or integer type. The input file may contain up to 366 days of hourly data with each day containing 24 hours. All dates must be in chronological order.

COLROW3D requires one input file. Table 4-9 is a sample input file. The input file contains two columns of data separated by at least one space. The first column is the date and time of the data expressed in decimal format, while the second column is the energy consumption data. When preparing the input file, keep in mind the following "Rules":

- *The input file may only contain numeric data of the integer and real type.* No characters other than the numerals 0 through 9, decimal points, minus signs, and spaces are allowed.

- *Each line row or record should contain only two data fields.* If more than two values are included, data beyond the second value are ignored. If only one datum is given on a line, the program will assume a missing value for the second field. A value of 0 is used as the missing code.

- *The maximum data that will be read are 366 days worth of hourly data.* Each day may contain from 1 to 24 hours of data--one record per hour. Only hourly data should be used as input to COLROW3D. Data in sub-hourly format must be converted to hourly format prior to processing.

- *The first column of data in the input file is the decimal date stamp.* The decimal date is a contrived method of representing the date and time using a single data string. Before you run COLROW3D, you must first convert your date and time to the decimal date format. The Princeton ARCHIVE program by Feuermann and Kempton (1987) is recommended for this purpose. A copy of ARCHIVE is included in this workbook.

Arbitrarily, January 1, 1980 00:00:00 hours is considered to be day 0 and has the decimal date representation 0000.0000. The number on the left hand side of the decimal point represents the number of days since January 1, 1980. The number on the right hand side of the decimal point represents the hour as a fraction of the day. Hours range from 0 through 23 and are calculated using the formula Hour = Decimal portion multiplied by 24 and rounded to the nearest integer. Hour 24 becomes Hour 0 of the following day. Note, the day of the year must be in chronological order. No such requirement is imposed on the hour of the day.

Valid dates are from January 1, 1980 (day 0) through December 31, 2009 (day 10957). Leap years and century leap years are taken into consideration. The program will need to be updated for decimal dates beyond the year 2009. Table 4-11 gives decimal dates for January 1 from 1980 through 2009. The following are examples of decimal date conversion

Date	**Time**	**Decimal date**
January 21, 1988	11 p.m.	2942.9583
May 1, 1990	1 a.m.	3773.0417
December 31, 1991	5 p.m.	4382.7083

- The second data column in the input file can be any consumption environmental data. Acceptable values are between -999.9 and 9999.9. A value of 0 will be used for missing data. If the value lies outside the acceptable range, the program records an error message to the .LOG file, and sets the hourly consumption to 0 for missing data. Data are recorded to the output file by rounding off to the first decimal place.

Examples of Energy use data

2901.0417 100 record indicates that on December 11, 1987 at 1:00 am the building used 100 kW of energy.

4020.0000 99999 ERROR! data value is out of bounds. A message will be written to the .LOG file, and the consumption will be set to 0.

Output file(s):

The output data file contains the original energy use data which have been rearranged in a matrix format for use with Lotus 123. This file must have a .3D extension. The .LOG file contains information written by COLROW3D while the program is executed. Information regarding date and time of run, and any errors encountered during processing are included. The date of the first and last string of processed data are shown.

Output File.

The output file is a N by 24 matrix containing only the valid input data. Here N stands for number of days between the first and last valid date stamp read from the input data file. For example, for one year's worth of data N is 366.

Both sample output files are shown in Table 4-9. The first row is a header that contains the hour of the day (ranging from 0 to 23), the first column is the day of the year (for example, day 121 is May 1st), and the remaining fields are hourly consumption data (in units of kWh/h). Missing data is represented by the value 0. The very first value in the first row shows the day of year for the last date read. This makes it convenient to use the output file in a spreadsheet since it can be used to compute the number of rows in the table.

.LOG File.

COLROW3D keeps a record of what happened during each run of the program. This information is written to disk in a .LOG file. The .LOG file has the same name as the input data file, but with a .LOG extension. Existing .LOG files with the same name will be overwritten. Any errors encountered during execution are written to the .LOG file. An example of a .LOG file is shown in Table 4-10.

The header specifies the name of the program and the date and time the run was made. The next line gives the name of the input file, the output file, the .LOG file, and the option selected. The following line gives the time the first record was read and the beginning date associated with that record.

The error table follows, and lists the location of the erroneous record, the data in the record, and the invalid datum. Since COLROW3D can deal with very large data files, a maximum of 50 date stamp errors and 20 data errors will be recorded in the .LOG file. This is to prevent a single bad datum from causing the entire data file to be written to the .LOG file. At the end of the error table is the time the last record was read and the ending date associated with that record.

The last part of the .LOG file consists of statistics about the input records. The .LOG file ends with a note, which states that the time portion of the date stamp is not checked for chronological order, and a message that the .LOG file is complete.

Execution:

COLROW3D *Input Output Option* <CR>

Input is the input file name (with complete path and extension specified.)

Output is the processed data file (with .3D extension).

Option is the integer 0 or 1. Choose 0 to create a file beginning with days = 1 and ending with day = 366, each day containing 24 hours of data. If the original file has fewer than 366 days of data, missing data are set to 0. Select a value of 1 to output fewer than one year's worth of data in contiguous order. Option 1 preserves the chronological order of input file. Figure 4-8 illustrates the difference in 3-D graphs between a "0" and "1" option. Table 4-9 shows the difference in the output files.

Example:

COLROW3D SAMPLE.DAT SAMPLE.3D 1 <CR>

Action: COLROW3D will read data from the input file called SAMPLE.DAT, output data to SAMPLE.3D, and create the .LOG file SAMPLE.LOG. Since the SAMPLE.DAT input file contains less than one year's worth of data, missing data are given the value of 0.

Example:

COLROW3D SAMPLE.DAT SAMPLE.3D 0 <CR>

Action: COLROW3D will read the file SAMPLE.DAT, output to the file SAMPLE.3D, and create the .LOG file SAMPLE.LOG. The output file is a 366 by 24 matrix with missing data set to 0.

Example:

COLROW3D ? <CR>

Action: COLROW3D displays the on-line help screen. For additional help, check the manual.

Notes:

(1) An additional program is available from the Energy Systems Lab that automatically creates the Lotus .PIC graphing instructions called 3DMAC.WK1. Feel free to contact the Energy Systems Laboratory about availability. This program is useful for automatically creating 3-D spreadsheet graphics in the batch mode.

Table 4-13 contains the 3DGRAPH plotting instructions that were used to produce the lower half at Figure 4-8. The plot instructions are also included in electronic form in the SAMPLEM1.3DP file that is included with the workbook diskette. After installing, configuring, and initiating the Intex 3DGRAPH Lotus add-on, the SAMPLE.3D data matrix can be loaded with a FILE IMPORT command (with the pointer in cell A1). The

SAMPLEM1.3DP graphing instructions can then be loaded after 3D Graph has been initiated with a GRAPH NAME USE command.

(2) COLROW3D will accept only numeric data of the real and integer type separated by at least one space. This program cannot handle any other data types. Two primary kinds of errors can occur: (i) Decimal date error, and ii) Energy use (or other data) error.

(i) Decimal date errors.

An error with the date stamp will occur if the day is not in chronological order. The hours of the day do not need to be in order. Example, if the input file reads:

3840.0000	24.2
3840.0417	25.3
3880.0000	28.0
3840.1250	25.9
3840.0833	25.5
....	...

The last two records will be skipped and reported as an error in the .LOG file because the decimal dates 3840.1250 and 3840.0833 are out of sequence with the record 3880.0000.

Another date error will occur if the decimal date value is less than 0 (January 1, 1980) or larger than 10957 (December 31, 2009). Refer to the COLROW3D manual for further details.

(ii) Energy use errors.

A data out of bounds error will occur if the consumption value lies outside the acceptable range [-999.9 ,9999.9]. If this should happen, the data record is skipped and an error message is written to the .LOG file. These limits are established by the maximum file import size in Lotus 123.

4.4 SUMMARY.

This section has been included to give the reader some helpful hints about what to do with the data once it has been collected from a logger. This section describes a collection of routines used to process and plot data collected from Synergistics loggers used by the Texas LoanSTAR Program over the last three years on a weekly basis. Instructions and sample code are provided for developing inspection and summary plots, and 3-D plots using a combination of public domain data processing toolkits and inexpensive commercially available plotting software.

TABLE 4-1 Example Synergistics raw data format 10192168.RAW.

```
6/ 9/92  0: 0: 0 391 "V  "    9.837   3.795   1.186   1.225   14.253   4.483   2.428   1.751   1.504   1.443  160.000
20.000 1180.000 7780.000   22.000
6/ 9/92  1: 0: 0 392 "V  "    9.905   3.746   1.213   1.256   13.551   4.461   2.417   1.540   1.541   1.515  144.000
0.000 1090.000 7230.000   22.000
6/ 9/92  2: 0: 0 393 "V  "   10.088   3.739   1.235   1.282   13.112   4.430   2.406   1.295   1.541   1.563  102.000
19.000  940.000 6480.000   22.000
6/ 9/92  3: 0: 0 394 "V  "   10.119   3.624   1.245   1.299   13.217   4.442   2.409   1.334   1.595   1.626  102.000
0.000 1000.000 6480.000   22.000
6/ 9/92  4: 0: 0 395 "V  "   10.239   3.619   1.249   1.301   13.436   4.508   2.420   1.290   1.612   1.647  100.000
21.000  940.000 6400.000   24.000
6/ 9/92  5: 0: 0 396 "V  "   10.302   3.619   1.253   1.313   13.562   4.536   2.419   1.241   1.620   1.662  102.000
21.000  930.000 6240.000   22.000
6/ 9/92  6: 0: 0 397 "V  "   10.365   3.642   1.250   1.316   13.572   4.571   2.427   1.233   1.610   1.659  102.000
21.000  940.000 6170.000   22.000
6/ 9/92  7: 0: 0 398 "V  "   10.365   3.821   1.223   1.298   13.499   4.803   2.475   1.259   1.571   1.628  114.000
0.000  920.000 6180.000   22.000
6/ 9/92  8: 0: 0 399 "V  "   10.177   3.732   1.203   1.278   13.918   4.728   2.504   0.758   1.615   1.670  130.000
19.000  960.000 6700.000   24.000
6/ 9/92  9: 0: 0 400 "V  "   10.574   3.605   1.240   1.315   16.911   4.596   2.515   0.439   1.680   1.713  196.000
41.000 1360.000 11170.000   24.000
6/ 9/92 10: 0: 0 401 "V  "   12.212   4.130   0.979   0.985   21.306   5.262   2.629   0.438   1.400   1.393  214.000
41.000 1310.000 9270.000   30.000
6/ 9/92 11: 0: 0 402 "V  "   12.474   4.756   1.089   1.132   25.240   4.596   2.720   0.436   1.304   1.236  210.000
0.000 1390.000 8850.000   32.000
6/ 9/92 12: 0: 0 403 "V  "   12.610   4.660   1.168   1.210   22.164   4.539   2.596   0.436   1.343   1.267  212.000
21.000 1390.000 8910.000   30.000
6/ 9/92 13: 0: 0 404 "V  "   13.353   3.857   1.236   1.287   20.782   4.508   2.510   0.436   1.474   1.406  214.000
20.000 1280.000 8440.000   30.000
6/ 9/92 14: 0: 0 405 "V  "   11.145   3.741   0.995   0.986   21.285   5.233   2.513   0.434   1.582   1.503  208.000
21.000 1240.000 7920.000   28.000
6/ 9/92 15: 0: 0 406 "V  "   10.145   3.857   0.934   0.905   21.641   5.648   2.502   0.433   1.614   1.534  206.000
20.000 1140.000 7600.000   30.000
6/ 9/92 16: 0: 0 407 "V  "    9.852   3.840   1.145   1.158   19.663   4.499   2.504   0.430   1.582   1.508  202.000
0.000 1040.000 6910.000   28.000
6/ 9/92 17: 0: 0 408 "V  "    9.727   3.727   1.002   0.971   15.163   5.243   2.521   0.995   1.614   1.530  198.000
21.000  870.000 5370.000   24.000
6/ 9/92 18: 0: 0 409 "V  "   10.030   3.751   1.118   1.112   14.598   4.543   2.527   1.312   1.605   1.538  196.000
21.000  940.000 5250.000   24.000
6/ 9/92 19: 0: 0 410 "V  "   10.224   4.127   1.076   1.059   15.341   4.511   2.709   1.380   1.523   1.490  200.000
0.000  930.000 5490.000   24.000
6/ 9/92 20: 0: 0 411 "V  "   10.025   4.229   1.059   1.074   15.037   4.552   2.755   1.976   1.435   1.434  202.000
22.000 1130.000 7140.000   24.000
6/ 9/92 21: 0: 0 412 "V  "   10.109   4.175   1.059   1.071   14.336   4.539   2.686   1.833   1.440   1.438  190.000
20.000 1050.000 6890.000   24.000
6/ 9/92 22: 0: 0 413 "V  "   10.088   4.973   1.052   1.058   14.117   4.527   2.690   1.700   1.309   1.274  194.000
21.000 1160.000 7270.000   24.000
6/ 9/92 23: 0: 0 414 "V  "   10.198   5.025   1.072   1.084   13.028   4.543   2.687   1.580   1.318   1.311  174.000
0.000 1140.000 7050.000   22.000
```

TABLE 4-2 *Example output 10192168.DAT from RAW2DAT program.*

```
6 9 92  0 0  9.837 3.795 1.186 1.225 14.253 4.483 2.428 1.751 1.504 1.443 160.000 20.000 1180.000 7780.000 22.000
6 9 92  1 0  9.905 3.746 1.213 1.256 13.551 4.461 2.417 1.540 1.541 1.515 144.000 0.000 1090.000 7230.000 22.000
6 9 92  2 0  10.088 3.739 1.235 1.282 13.112 4.430 2.406 1.295 1.541 1.563 102.000 19.000 940.000 6480.000 22.000
6 9 92  3 0  10.119 3.624 1.245 1.299 13.217 4.442 2.409 1.334 1.595 1.626 102.000 0.000 1000.000 6480.000 22.000
6 9 92  4 0  10.239 3.619 1.249 1.301 13.436 4.508 2.420 1.290 1.612 1.647 100.000 21.000 940.000 6400.000 24.000
6 9 92  5 0  10.302 3.619 1.253 1.313 13.562 4.536 2.419 1.241 1.620 1.662 102.000 21.000 930.000 6240.000 22.000
6 9 92  6 0  10.365 3.642 1.250 1.316 13.572 4.571 2.427 1.233 1.610 1.659 102.000 21.000 940.000 6170.000 22.000
6 9 92  7 0  10.365 3.821 1.223 1.298 13.499 4.803 2.475 1.259 1.571 1.628 114.000 0.000 920.000 6180.000 22.000
6 9 92  8 0  10.177 3.732 1.203 1.278 13.918 4.728 2.504 0.758 1.615 1.670 130.000 19.000 960.000 6700.000 24.000
6 9 92  9 0  10.574 3.605 1.240 1.315 16.911 4.596 2.515 0.439 1.680 1.713 196.000 41.000 1360.000 11170.000 24.000
6 9 92  10 0  12.212 4.130 0.979 0.985 21.306 5.262 2.629 0.438 1.400 1.393 214.000 41.000 1310.000 9270.000 30.000
6 9 92  11 0  12.474 4.756 1.089 1.132 25.240 4.596 2.720 0.436 1.304 1.236 210.000 0.000 1390.000 8850.000 32.000
6 9 92  12 0  12.610 4.660 1.168 1.210 22.164 4.539 2.596 0.436 1.343 1.267 212.000 21.000 1390.000 8910.000 30.000
6 9 92  13 0  13.353 3.857 1.236 1.287 20.782 4.508 2.510 0.436 1.474 1.406 214.000 20.000 1280.000 8440.000 30.000
6 9 92  14 0  11.145 3.741 0.995 0.986 21.285 5.233 2.513 0.434 1.582 1.503 208.000 21.000 1240.000 7920.000 28.000
6 9 92  15 0  10.145 3.857 0.934 0.905 21.641 5.648 2.502 0.433 1.614 1.534 206.000 20.000 1140.000 7600.000 30.000
6 9 92  16 0  9.852 3.840 1.145 1.158 19.663 4.499 2.504 0.430 1.582 1.508 202.000 0.000 1040.000 6910.000 28.000
6 9 92  17 0  9.727 3.727 1.002 0.971 15.163 5.243 2.521 0.995 1.614 1.530 198.000 21.000 870.000 5370.000 24.000
6 9 92  18 0  10.030 3.751 1.118 1.112 14.598 4.543 2.527 1.312 1.605 1.538 196.000 21.000 940.000 5250.000 24.000
6 9 92  19 0  10.224 4.127 1.076 1.059 15.341 4.511 2.709 1.380 1.523 1.490 200.000 0.000 930.000 5490.000 24.000
6 9 92  20 0  10.025 4.229 1.059 1.074 15.037 4.552 2.755 1.976 1.435 1.434 202.000 22.000 1130.000 7140.000 24.000
6 9 92  21 0  10.109 4.175 1.059 1.071 14.336 4.539 2.686 1.833 1.440 1.438 190.000 20.000 1050.000 6890.000 24.000
6 9 92  22 0  10.088 4.973 1.052 1.058 14.117 4.527 2.690 1.700 1.309 1.274 194.000 21.000 1160.000 7270.000 24.000
6 9 92  23 0  10.198 5.025 1.072 1.084 13.028 4.543 2.687 1.580 1.318 1.311 174.000 0.000 1140.000 7050.000 22.000
```

TABLE 4-3 Example channel table for the 10190001.CHT ARCHIVE program.

Date MM/DD/YY (YY DDD)	Time HH:mm	Raw-Data lin pos	coln pos	Arch coln pos	Name of Channel	Archive Units	Arch Format	Conv'n Code	Conv'n Constants	Error Code	Error Constants	Channel Description
#												
04/23/92	00:00	1	0	0	Begin	UTC						Beginning dat
04/23/92	00:00	1	1	1	Bldg. #	xx	I3	2	0 101	0		*Good startin
04/23/92	00:00	1	1	2	Mon-Raw	MM	I3	1		0		Month
04/23/92	00:00	1	2	3	Mon-Raw	DD	I3	1		0		Day
04/23/92	00:00	1	3	4	Mon-Raw	YY	I3	1		0		Year
04/23/92	00:00	1	3	5	Greg-Jul	MMDDYY	I5	24	1 2	0		Gregorian Dat
04/23/92	00:00	1	4	7	Time	HH mm	I5	16	5	0		Time
04/23/92	00:00	1	3	6	Greg-Dec	DDD.frac	F10.4	28		0		Gregorian Dat
04/23/92	00:00	1	6	8	S/RMCC1	F9.3	F9.3	1		1	-5 500	S/R MCC 1 (kW
04/23/92	00:00	1	7	9	S/RMCC1	F9.3	F9.3	1		1	-5 500	S/R AHUs 1s,2
04/23/92	00:00	1	8	19	HDF1-2N1	F9.3	F9.3	1		1	-5 500	Hot Deck Fans
04/23/92	00:00	1	9	20	HDF1-2N2	F9.3	F9.3	1		1	-5 500	Hot Deck Fans
04/23/92	00:00	1	10	10	S/RMCC2	F9.3	F9.3	1		1	-5 500	S/R MCC 2 (kW
04/23/92	00:00	1	11	11	S/RMC2/2	F9.3	F9.3	1		1	-5 500	S/R AHU 2n, R
04/23/92	00:00	1	12	12	S/RMC2/1	F9.3	F9.3	1		1	-5 500	S/R AHU 1n, R
04/23/92	00:00	1	13	13	ChWPump	F9.3	F9.3	1		1	0 500	ChW Pump (kWh
04/23/92	00:00	1	14	21	HDF1-2S1	F9.3	F9.3	1		1	-5 500	Hot Deck Fans
04/23/92	00:00	1	15	22	HDF1-2S2	F9.3	F9.3	1		1	-5 500	Hot Deck Fans
04/23/92	00:00	1	16	14	kWh M A	F9.3	F9.3	1		1	0 500	Bldg kWh Mete
04/23/92	00:00	1	17	15	CondRet	F9.3	F9.3	1		1	0 99999	Cond Ret Mete
04/23/92	00:00	1	18	16	ChWBtuT	F9.3	F9.3	1		1	0 99999	ChW Btu (kBtu
04/23/92	00:00	1	19	17	ChWGalT	F9.3	F9.3	1		1	0 99999	ChW Gal (gal)
04/23/92	00:00	1	20	18	kWh M B	F9.3	F9.3	1		1	0 500	Bldg kWh Mete
04/11/99	23:00	1	0	0	End	UTC						

TABLE 4-4 *Example .LOG file from the ARCHIVE program.*

Log of Archive, version: 1.41 of 15 June 1987, processed on 24 Jun 1992

Files:

RAW DATA 10192168.dat
CHANNEL TABLE 10190001.cht
ARCHIVE 10192168.ach
LOG 10192168.log

Archive delimiter is " ".
Missing or bad data values are replaced by the value -99.000 .

Line errors: are identified by their line number in the raw data file.
Data errors: are identified by the channels name, line and position
within the case: "name "(line in case/position in line).
Line numbers in raw data file are shown as |number| or as *number*
|numbers| indicates a line of data, *numbers* is a comment line.

First case on raw data: 92 161 00:00

BeginDate: 92 114 00:00 First output case: 92 161 00:00

- -

- -

EndDate: 99 101 23:00 Last output case: 92 167 23:00

STATISTICS:

168 lines read from beginning of raw data file.
168 lines processed between Begin and End dates.
(including 0 comments and 0 all-blank lines)
0 line errors detected.

0 data errors, and 0 missing data detected

***TABLE 4-5** Summary of GRAPHER instructions for graph T1017.GRF.*

	AXIS X	AXIS Y	DATA FILENMS		COLUMNS X	COLUMNS Y	LINE	CENTERED SYM.	BEST FIT
TYPE:	LINEAR	LINEAR	1	T101	F	N	SOLID	NO	NO
TITLE:	Site 101 Building Meter 2								
START:	1.5,1.0	1.5,1.0	3						
LENGTH:	6.0	6.0	4						
START:	4543.0	0.0	5						
END:	4550.0	0.0	6						
TICS:	YES	YES	7						
TIC LAB:	YES	YES	8						
AXIS FILE:	X-AXIS	Y-AXIS							
GRID FILE:									
TEXT FILE:									
DIV.FILE:									

Table 4-6 *T1017.GRF GRAPHER file.*

```
1243
1 2 0 0 0
t101
70 78 48 19"NO " 48
"YES" "SOLID" 1.500e-001 1
"NO" 41 1.000e-001 1 1
48 9.900e+028 9.900e+028 0.000e+000 "DEFAULT" 1.000e-001 1
"SOLID" 0 1.500e-001 9.9000000e+029 9.9000000e+029 200 2.000e+000 1
9.9000000e+028 9.9000000e+029 9.9000000e+028 9.9000000e+029 1.500e-001
X-AXIS
1.5000000e+000 1.0000000e+000 6.0000000e+000 88
3.9200000e+003 3.9270000e+003 9.9000000e+028 1 1
0.0000000e+000 1.0000000e+000 1.5000000e-001 1 1
7 0 1
1 9.9000000e+028 0.0000000e+000 9.9000000e+028 2.5000000e-001
"DEFAULT" "DEFAULT" "Site 101 Beginning"
4.0000000e-002
Y-AXIS
1.5000000e+000 1.0000000e+000 6.0000000e+000 89
0.0000000e+000 9.9000000e+028 9.9000000e+028 1 1
2.7000000e+002 9.9000000e+028 1.5000000e-001 1 1
10 1 1
1 9.9000000e+028 0.0000000e+000 9.9000000e+028 2.5000000e-001
"DEFAULT" "DEFAULT" "Bldg Meter A (kWh/h)"
4.0000000e-002
```

***TABLE 4-7** Example output from the ARCHIVE program, file 10192168.ACH.*

```
101 6 9 92 92161 4543.0000 0 9.837 3.795 14.253 4.483 2.428 1.751 160.000 20.000 1180.000
7780.000 22.000 1.186 1.225 1.504 1.443
101 6 9 92 92161 4543.0417 100 9.905 3.746 13.551 4.461 2.417 1.540 144.000 0.000 1090.000
7230.000 22.000 1.213 1.256 1.541 1.515
101 6 9 92 92161 4543.0833 200 10.088 3.739 13.112 4.430 2.406 1.295 102.000 19.000 940.000
6480.000 22.000 1.235 1.282 1.541 1.563
101 6 9 92 92161 4543.1250 300 10.119 3.624 13.217 4.442 2.409 1.334 102.000 0.000 1000.000
6480.000 22.000 1.245 1.299 1.595 1.626
101 6 9 92 92161 4543.1667 400 10.239 3.619 13.436 4.508 2.420 1.290 100.000 21.000 940.000
6400.000 24.000 1.249 1.301 1.612 1.647
101 6 9 92 92161 4543.2083 500 10.302 3.619 13.562 4.536 2.419 1.241 102.000 21.000 930.000
6240.000 22.000 1.253 1.313 1.620 1.662
101 6 9 92 92161 4543.2500 600 10.365 3.642 13.572 4.571 2.427 1.233 102.000 21.000 940.000
6170.000 22.000 1.250 1.316 1.610 1.659
101 6 9 92 92161 4543.2917 700 10.365 3.821 13.499 4.803 2.475 1.259 114.000 0.000 920.000
6180.000 22.000 1.223 1.298 1.571 1.628
101 6 9 92 92161 4543.3333 800 10.177 3.732 13.918 4.728 2.504 0.758 130.000 19.000 960.000
6700.000 24.000 1.203 1.278 1.615 1.670
101 6 9 92 92161 4543.3750 900 10.574 3.605 16.911 4.596 2.515 0.439 196.000 41.000 1360.000
11170.000 24.000 1.240 1.315 1.680 1.713
101 6 9 92 92161 4543.4167 1000 12.212 4.130 21.306 5.262 2.629 0.438 214.000 41.000 1310.000
9270.000 30.000 0.979 0.985 1.400 1.393
101 6 9 92 92161 4543.4583 1100 12.474 4.756 25.240 4.596 2.720 0.436 210.000 0.000 1390.000
8850.000 32.000 1.089 1.132 1.304 1.236
101 6 9 92 92161 4543.5000 1200 12.610 4.660 22.164 4.539 2.596 0.436 212.000 21.000 1390.000
8910.000 30.000 1.168 1.210 1.343 1.267
101 6 9 92 92161 4543.5417 1300 13.353 3.857 20.782 4.508 2.510 0.436 214.000 20.000 1280.000
8440.000 30.000 1.236 1.287 1.474 1.406
101 6 9 92 92161 4543.5833 1400 11.145 3.741 21.285 5.233 2.513 0.434 208.000 21.000 1240.000
7920.000 28.000 0.995 0.986 1.582 1.503
101 6 9 92 92161 4543.6250 1500 10.145 3.857 21.641 5.648 2.502 0.433 206.000 20.000 1140.000
7600.000 30.000 0.934 0.905 1.614 1.534
101 6 9 92 92161 4543.6667 1600 9.852 3.840 19.663 4.499 2.504 0.430 202.000 0.000 1040.000
6910.000 28.000 1.145 1.158 1.582 1.508
101 6 9 92 92161 4543.7083 1700 9.727 3.727 15.163 5.243 2.521 0.995 198.000 21.000 870.000
5370.000 24.000 1.002 0.971 1.614 1.530
101 6 9 92 92161 4543.7500 1800 10.030 3.751 14.598 4.543 2.527 1.312 196.000 21.000 940.000
5250.000 24.000 1.118 1.112 1.605 1.538
101 6 9 92 92161 4543.7917 1900 10.224 4.127 15.341 4.511 2.709 1.380 200.000 0.000 930.000
5490.000 24.000 1.076 1.059 1.523 1.490
101 6 9 92 92161 4543.8333 2000 10.025 4.229 15.037 4.552 2.755 1.976 202.000 22.000 1130.000
7140.000 24.000 1.059 1.074 1.435 1.434
101 6 9 92 92161 4543.8750 2100 10.109 4.175 14.336 4.539 2.686 1.833 190.000 20.000 1050.000
6890.000 24.000 1.059 1.071 1.440 1.438
101 6 9 92 92161 4543.9167 2200 10.088 4.973 14.117 4.527 2.690 1.700 194.000 21.000 1160.000
7270.000 24.000 1.052 1.058 1.309 1.274
101 6 9 92 92161 4543.9583 2300 10.198 5.025 13.028 4.543 2.687 1.580 174.000 0.000 1140.000
7050.000 22.000 1.072 1.084 1.318 1.311
```

TABLE 4-8 *Example input data file for COLROW3D.*

An Example of An Input Data File

4138.0000	928.74
4138.0417	915.18
4138.0833	903.13
4138.1250	876.25
4138.1667	864.20
4138.2083	857.17
4138.2500	861.44
4138.2917	883.79
4138.3333	982.74

TABLE 4-9 *An example output data file for COLROW3D.*

Method "0"

366	0	1	2	3	4		20	21	22	23
1	929	915	903	876	864	...	980	982	984	957
2	910	883	869	855	840	...	969	974	988	968
					.					
					.					
					.					
366	918	887	863	850	837	...	877	889	895	892

Method "1"

120	0	1	2	3	4	...	20	21	22	23
359	831	828	822	816	809	...	839	853	850	851
360	839	822	813	811	809	...	835	846	853	844
				.						
				.						
				.						
366	877	873	864	874	879	...	874	889	877	883
1	875	851	811	807	807	...	859	863	867	864
2	351	839	831	827	818	...	821	832	839	841
3	841	831	827	825	825	...	857	866	855	852
				.						
				.						
				.						
120	894	883	881	878	872	...	926	921	922	920

TABLE 4-10 *An example .LOG file for COLROW3D.*

Sample .LOG file.

```
.LOG of Colrow3D run Fri Aug 16 01:59:30 1991
Raw data file        : njb0691.DAT
Colrow3D matrix file  : NJB0691.3D
.LOG file            : NJB0691..LOG
Method used          : 1
First record read at 01:59:31     Begin Date : 4138.0000
        The following records were skipped
Record    Decimal Date     kWh/h data    Incorrect Value
-------------------------------------------------------------
  9      4138.3335        99982.7422        kwh
 49      4139.9165          987.5110        date
-------------------------------------------------------------
Last record read at 01:59:55      End Date : 4198.9585
Statistics :
 Total number of records read:              1464
 Total number of records processed:                1462
 Total number of records skipped:              2
 Total number of bad Decimal Dates:            1
 Total number of bad data values:              1

Notice : Time values within a day are NOT checked for
chrono.LOGical order.
        *** Error report completed. ***
```

***TABLE 4-11** Decimal Date Reference Table for COLROW3D. The following is a table of decimal dates for January 1 for the years from 1980 through 2009.*

	Year	Dec.Date	# Days
January 1	1981	366	365
January 1	1982	731	365
January 1	1983	1096	365
January 1	1984	1461	366
January 1	1985	1827	365
January 1	1986	2192	365
January 1	1987	2557	365
January 1	1988	2922	366
January 1	1989	3288	365
January 1	1990	3653	365
January 1	1991	4018	365
January 1	1992	4383	366
January 1	1993	4749	365
January 1	1994	5114	365
January 1	1995	5479	365
January 1	1996	5844	366
January 1	1997	6210	365
January 1	1998	6575	365
January 1	1999	6940	365
January 1	2000	7305	366
January 1	2001	7671	365
January 1	2002	8036	365
January 1	2003	8401	365
January 1	2004	8766	366
January 1	2005	9132	365
January 1	2006	9497	365
January 1	2007	9862	365
January 1	2008	10227	366
January 1	2009	10593	365

TABLE 4-12 Files included with the distribution diskette.

SAMPLEM0	3D	64961	09-24-91	10:50a
SAMPLEM1	3D	22481	09-24-91	10:50a
SAMPLEM1	3DP	512	07-11-92	1:29p
10192168	ACH	31920	06-24-92	9:32a
10192168	ACS	31920	06-24-92	9:32a
101CHGRF	AWK	145	10-18-90	11:08p
101DATE	AWK	443	06-19-92	10:19a
101SUMM	AWK	639	05-06-92	12:22p
RAW2DAT	AWK	4142	06-27-91	2:38p
101CHGRF	BAT	72	07-26-92	4:26p
101GRAPH	BAT	1763	07-26-92	4:22p
R2A	BAT	2768	06-19-92	10:21a
UTSUMM	BAT	1614	06-19-92	10:38a
10190001	CHT	3053	05-06-92	11:54a
ARCHIVE	COM	40937	06-16-87	5:07p
COLS	COM	13551	06-16-87	10:55a
ABUT	COM	13381	06-16-87	10:58a
DAYDAT	COM	20210	08-31-87	12:17a
KDOW	COM	20058	08-31-87	12:19a
KEEP	COM	20321	08-31-87	12:13a
QSELECT	COM	20138	08-31-87	12:15a
REPL	COM	15377	06-16-87	10:56a
ROWS	COM	27259	06-16-87	10:55a
SELECT	COM	20011	08-31-87	12:17a
TAIL	COM	12656	06-16-87	10:58a
TIMERGE	COM	20661	08-31-87	12:16a
TOTAL	COM	24710	08-31-87	12:19a
WDOW	COM	20058	08-31-87	12:20a
WEED	COM	20351	08-31-87	12:18a
10192168	DAT	19891	06-24-92	9:32a
SAMPLE	DAT	57430	09-20-91	8:58a
T101	DAT	31920	06-24-92	9:32a
DIR	DIR	0	08-18-92	11:20a
ABUT	DOC	1976	04-24-87	1:09p
ARTTOOL	DOC	5376	08-31-87	2:01a
COLS	DOC	1921	04-22-87	9:36p
DAYDAT	DOC	1920	08-17-57	6:00p
KDOW	DOC	1280	08-17-57	6:01p
KEEP	DOC	1024	08-31-87	12:46a
QSELECT	DOC	1024	08-31-87	12:48a
REPL	DOC	4470	04-23-87	5:23p
ROWS	DOC	13420	06-17-87	11:46a
SELECT	DOC	1536	08-31-87	12:50a
TAIL	DOC	1517	04-24-87	1:18p
TIMERGE	DOC	2560	08-31-87	2:05a
TOOLBOX	DOC	2703	06-17-87	11:42a
TOTAL	DOC	1792	08-31-87	2:10a
WDOW	DOC	640	08-17-87	5:43p
WEED	DOC	768	08-31-87	12:52a
COLROW3D	EXE	92787	09-19-91	8:54a
GAWK	EXE	134446	02-25-90	9:32p
MISSING	EXE	37471	11-13-91	4:27p
T1011	GRF	870	04-09-91	1:15p
T10110	GRF	865	04-09-91	1:15p
T10111	GRF	870	04-09-91	1:15p
T10112	GRF	870	05-06-92	12:17p
T10113	GRF	870	05-06-92	12:17p
T10114	GRF	870	05-06-92	12:17p
T10115	GRF	870	05-06-92	12:18p
T1012	GRF	867	04-09-91	1:15p
T1013	GRF	870	04-09-91	1:16p
T1014	GRF	864	04-09-91	1:16p
T1015	GRF	864	04-09-91	1:16p
T1016	GRF	867	04-09-91	1:16p
T1017	GRF	870	04-09-91	1:16p
T1018	GRF	874	04-09-91	1:16p
T1019	GRF	865	04-09-91	1:17p
TS1011	GRF	857	12-16-90	2:05a
TS10112	GRF	867	12-16-90	12:55a
TS1012	GRF	864	12-16-90	12:54a
TS1013	GRF	869	02-22-91	8:54a
TS1014	GRF	875	01-11-91	11:03a
TS1015	GRF	875	01-11-91	11:03a
TS1016	GRF	870	12-16-90	10:00p
TS1017	GRF	870	05-06-91	3:14p
TS1018	GRF	869	05-06-91	3:14p
TS1019	GRF	868	12-16-90	10:01p
10192168	LOG	1647	06-24-92	9:32a
MISSING	LOG	37	06-24-92	9:32a
SAMPLEM0	LOG	763	09-24-91	10:50a
101ONE	OUT	128275	06-24-92	9:35a
101SUMM	OUT	144613	06-24-92	9:43a
101TWO	OUT	31428	06-24-92	9:35a
ABUT	PAS	5601	06-16-87	10:58a
ARCHIVE	PAS	15869	06-16-87	5:09p
ARCPROC0	PAS	11410	06-10-87	3:19p
ARCPROC1	PAS	50829	06-15-87	11:40a
ARCPROC2	PAS	14038	06-09-87	5:10p
COLS	PAS	6064	06-16-87	10:55a
DATAUTIL	PAS	8758	08-20-87	4:17p
DAYDAT	PAS	2165	07-30-87	5:24p
KDOW	PAS	2432	07-30-87	5:07p
KEEP	PAS	1690	08-31-87	12:11a
QSELECT	PAS	2304	07-30-87	5:08p
REPL	PAS	9668	06-16-87	10:56a
ROWS	PAS	34515	06-16-87	10:54a
SELECT	PAS	2048	07-30-87	5:06p
TAIL	PAS	3652	06-16-87	10:18a
TIMERGE	PAS	3200	07-30-87	5:09p
TOTAL	PAS	8770	07-30-87	5:45p
WDOW	PAS	2432	07-30-87	5:10p
WEED	PAS	1792	07-30-87	5:11p
WRAP	PAS	14536	08-20-87	4:16p
SAMPLEM0	PIC	118867	09-24-91	11:07a
SAMPLEM1	PIC	44239	09-24-91	11:05a
A	PLT	21	07-18-90	1:30p
B	PLT	512	11-16-89	10:24p
C	PLT	512	11-16-89	10:51p
T1017	PLT	0	07-26-92	4:16p
10192168	RAW	31248	06-16-92	3:03p
20292168	WEA	10224	06-17-92	12:24p

113 file(s) 1629853 bytes

TABLE 4-13 *Intex solutions 3DGRAPH plotting instructions for 3D Surface plot SAMPLEM1.3DP.*

```
TYPE SURFACE HIDDEN
X  B1..Y1
Y  A2..A127
A  B2..Y127
OPTIONS TITLE     FIRST "EXAMPLE PLOT FOR COLROW3D"
                  SECOND "INPUT=SAMPLE.DAT, OPTION=1"
                  X AXIS  "HOUR OF DAY"
                  Y AXIS  "DAY OF YEAR"
                  Z AXIS  "ELECTRICITY USE (KWH/H)"
OPTIONS SCALE     Z-SCALE AUTOMATIC
                  X-SKIP 2
                  Y-SKIP 15
OPTIONS B&W
DISPLAY ROTATION 270
          VIEWPOINT MEDIUM
          AXIS YES
```

***FIGURE 4.1a** Flow chart for the GRAPHER program.* This figure illustrates the basic command flow chart for the GRAPHER software.

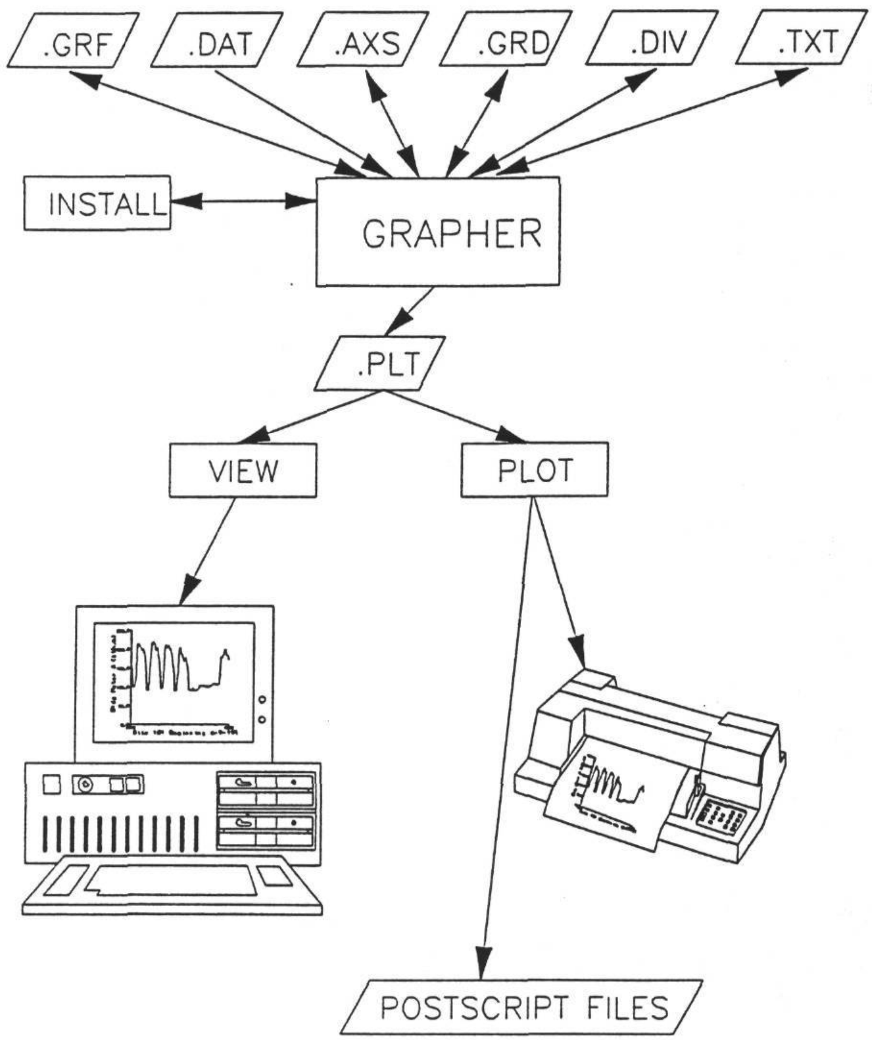


FIGURE 4.1b *Graph created with T1017.GRF and T101.DAT data file.*

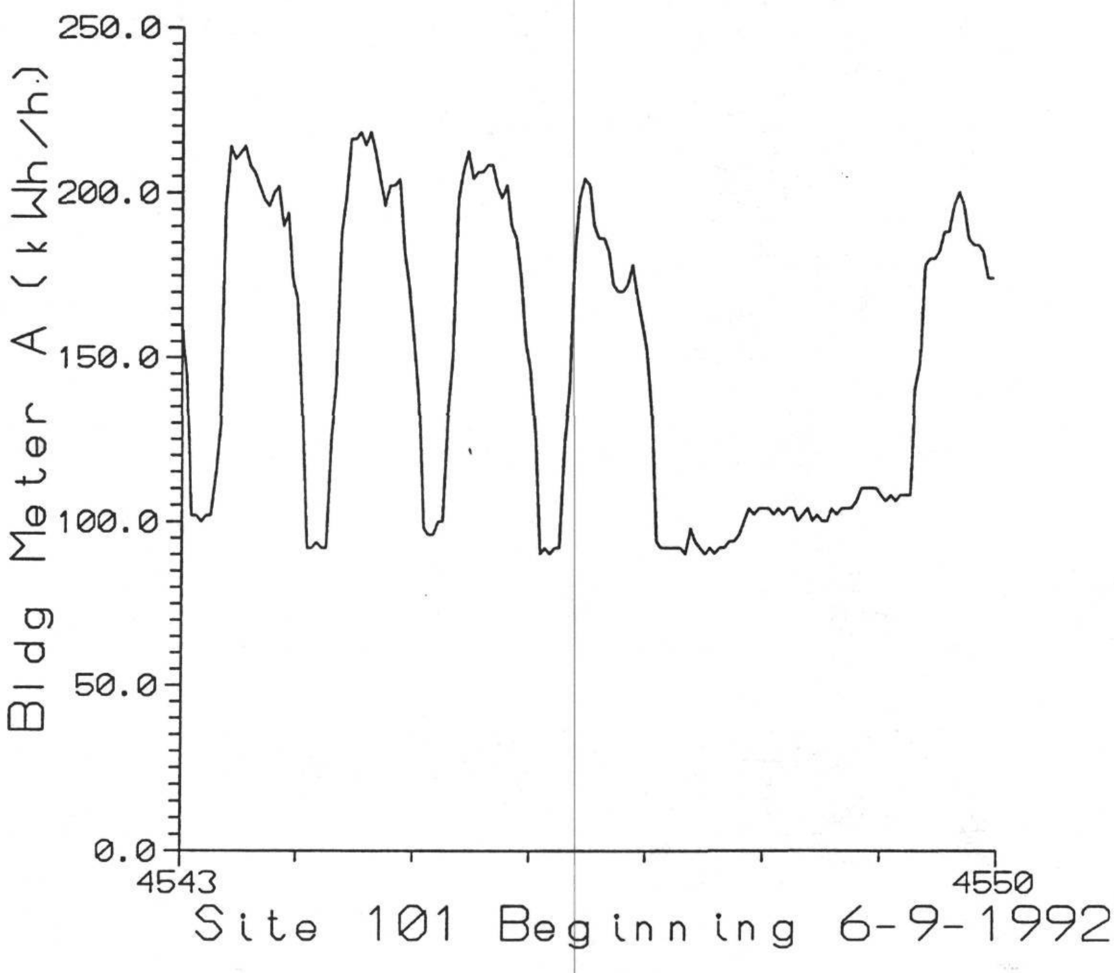


FIGURE 4.2a Example plot of the first 12 channels from site 101.

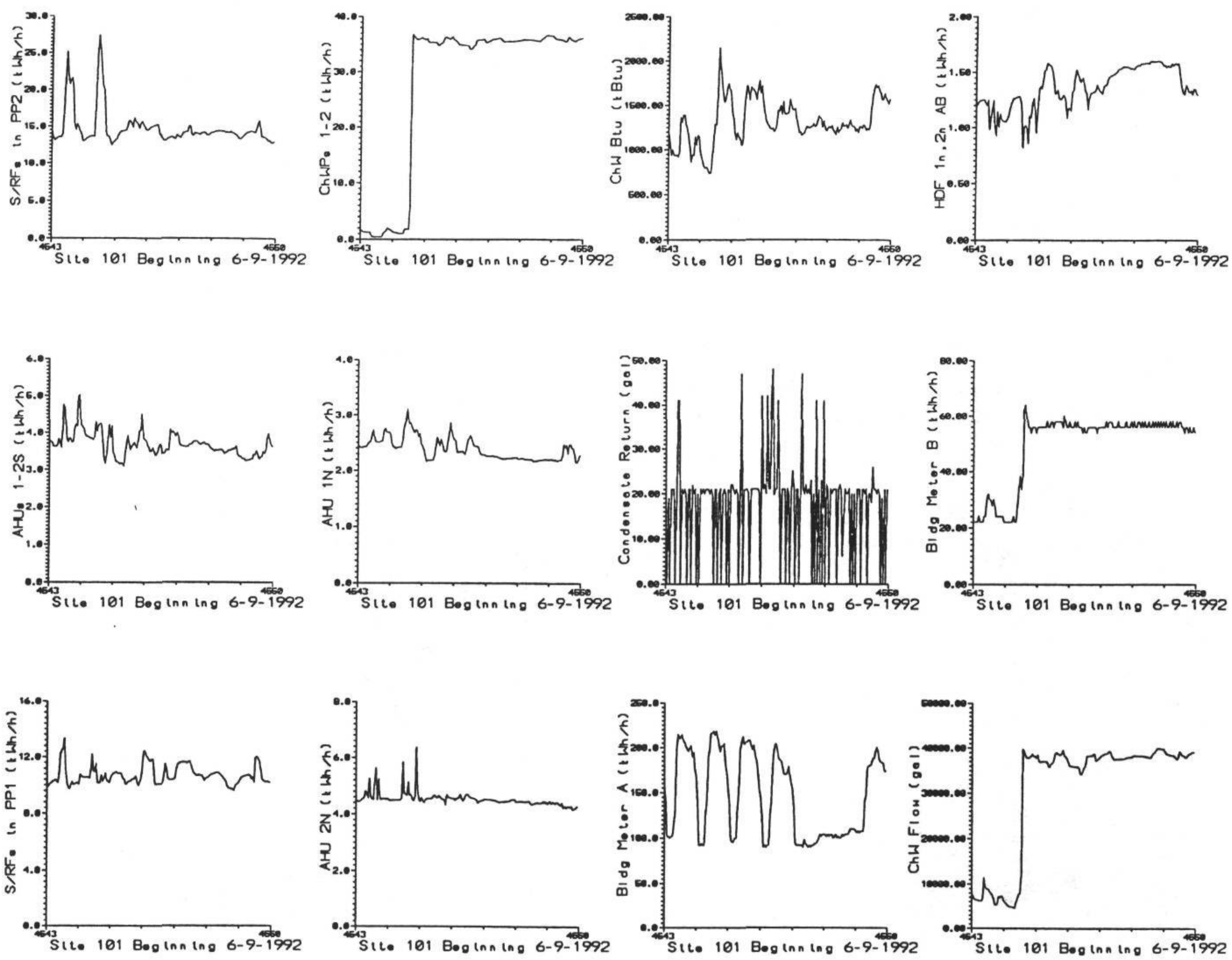


FIGURE 4.2b Example plot of the remaining channels for site 101.

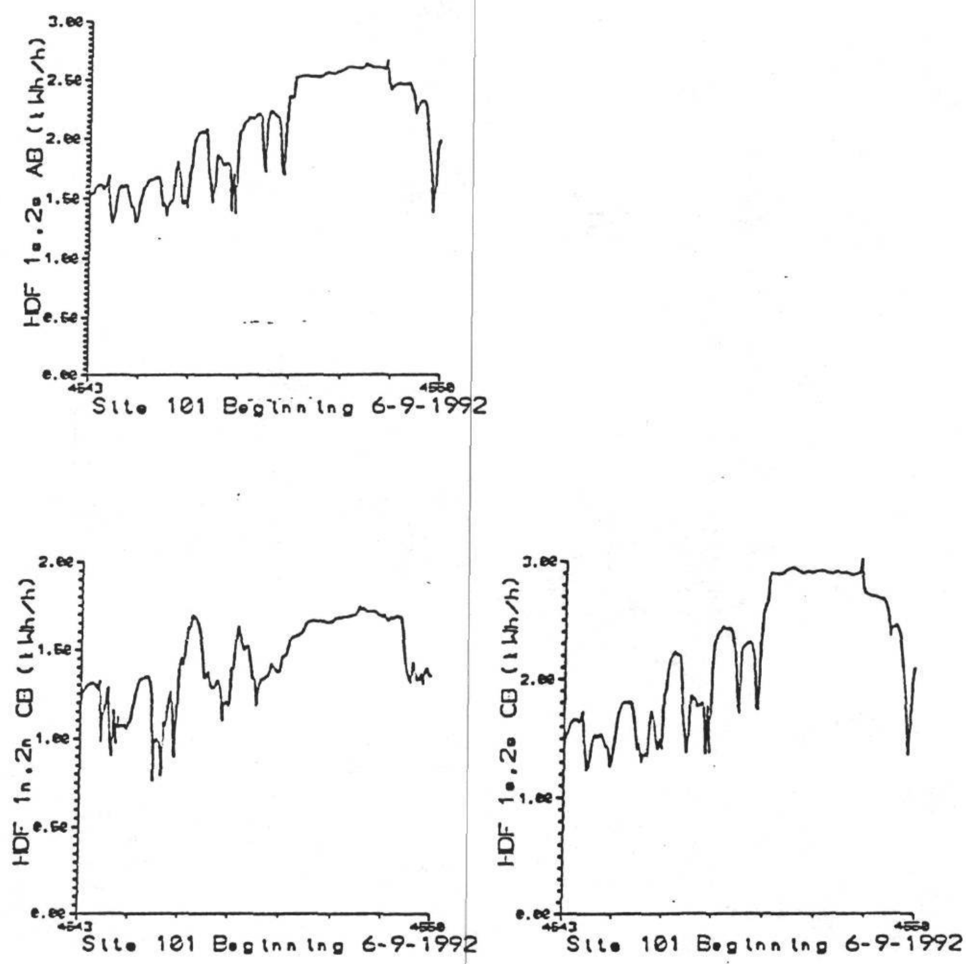


***FIGURE 4.3** Flow chart for R2A.BAT plotting procedure.*

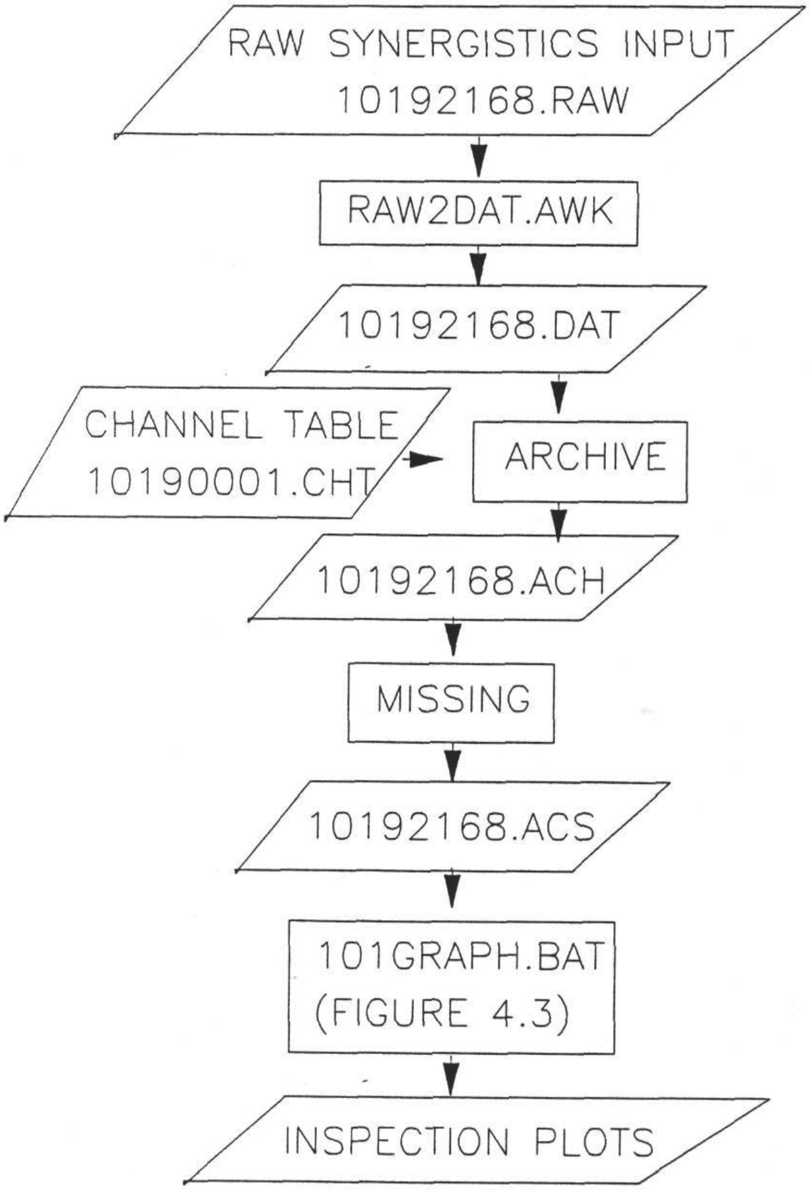


FIGURE 4.4 Flow chart for 101GRAPH.BAT.

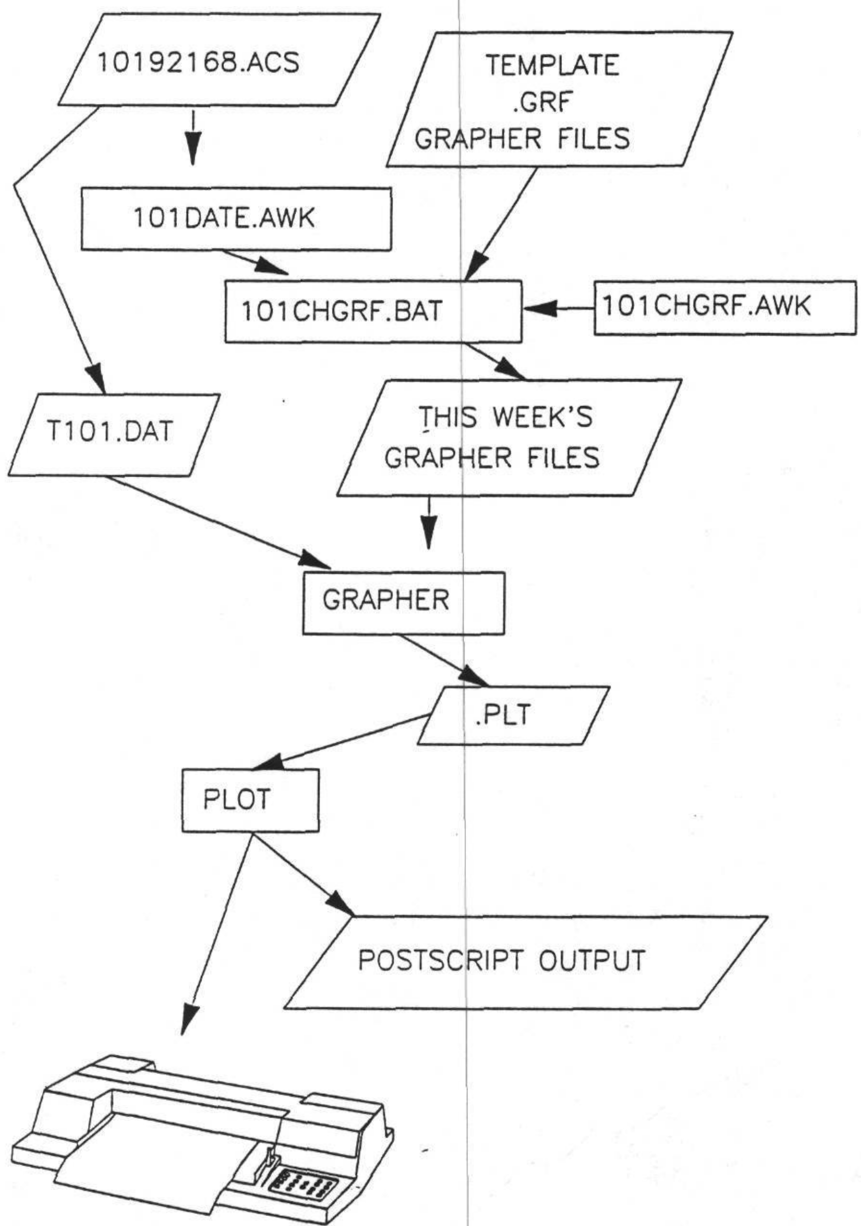


FIGURE 4.5 Example summary plot for site 101.

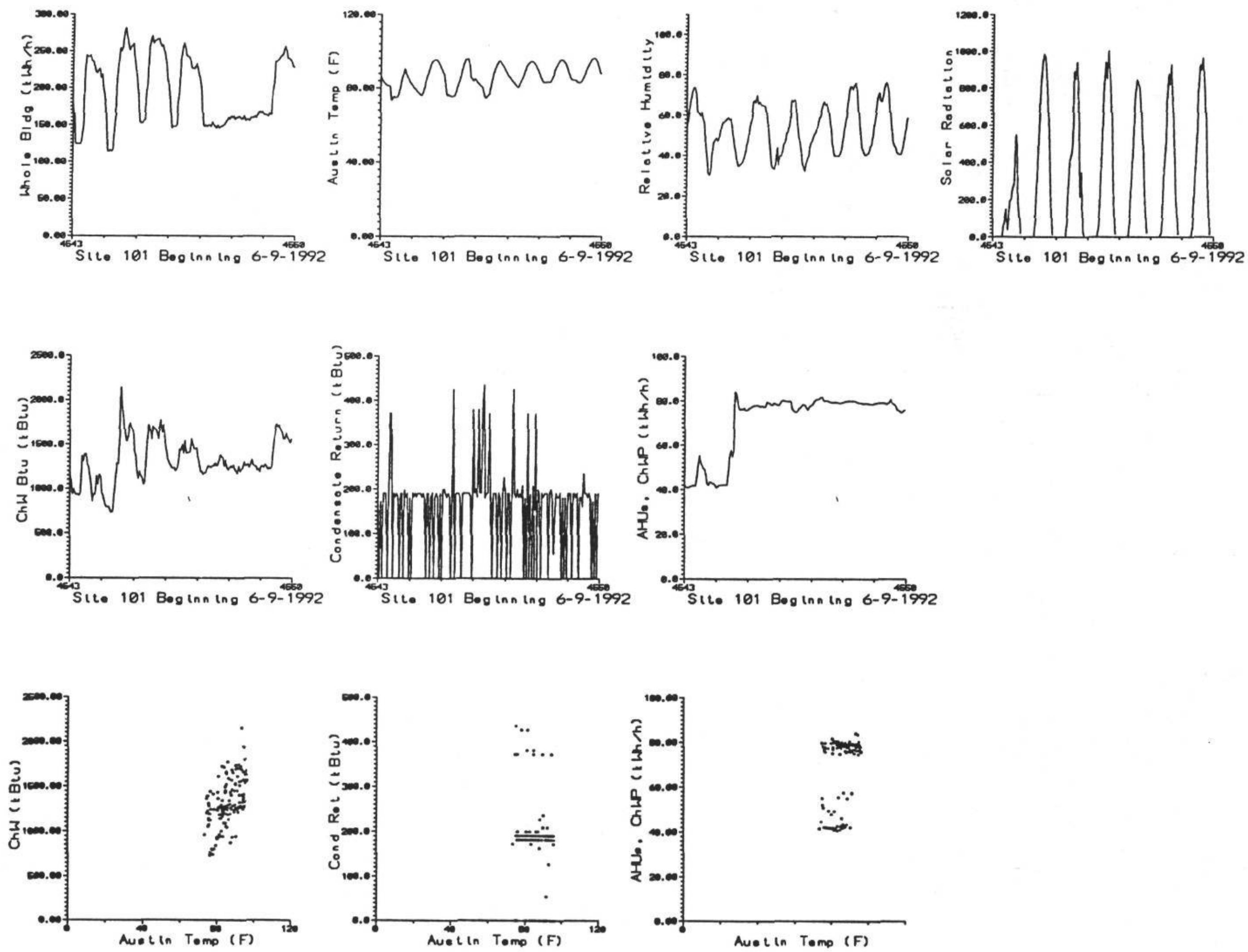


***FIGURE 4.6** Flow chart for summary page UTSUMM.BAT.*

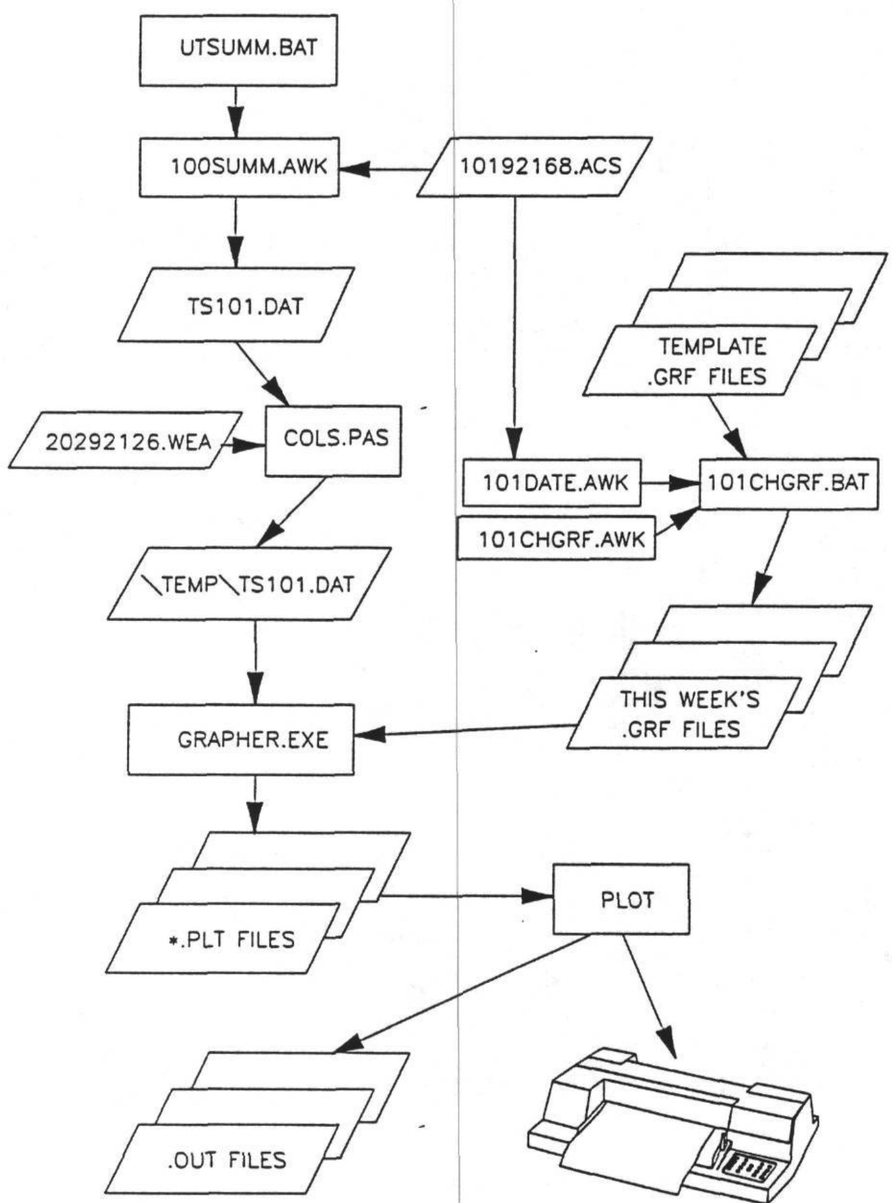


***FIGURE 4-7** Basic flow chart for producing 3-D plots.* This figure shows a basic flow chart for producing 3-D plots using several commercially available software packages and data processing routines from the LoanSTAR program.

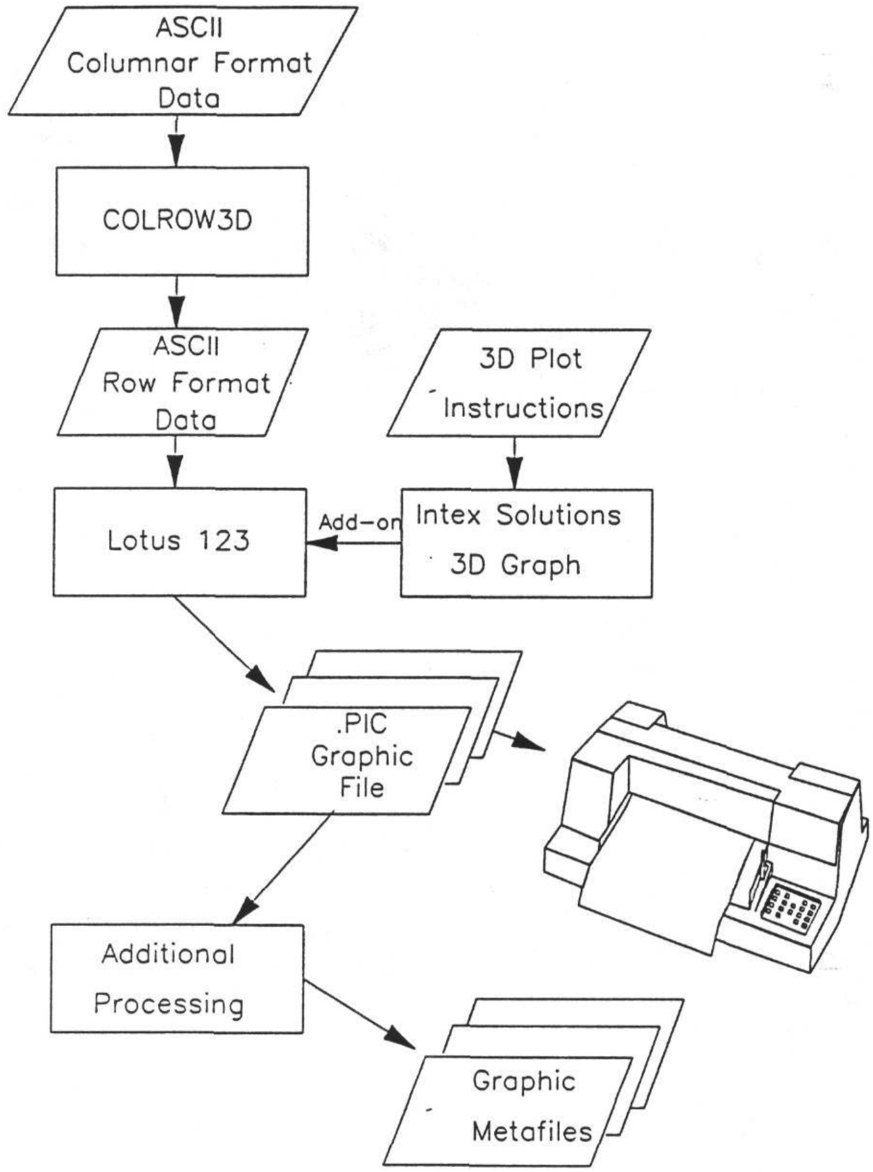
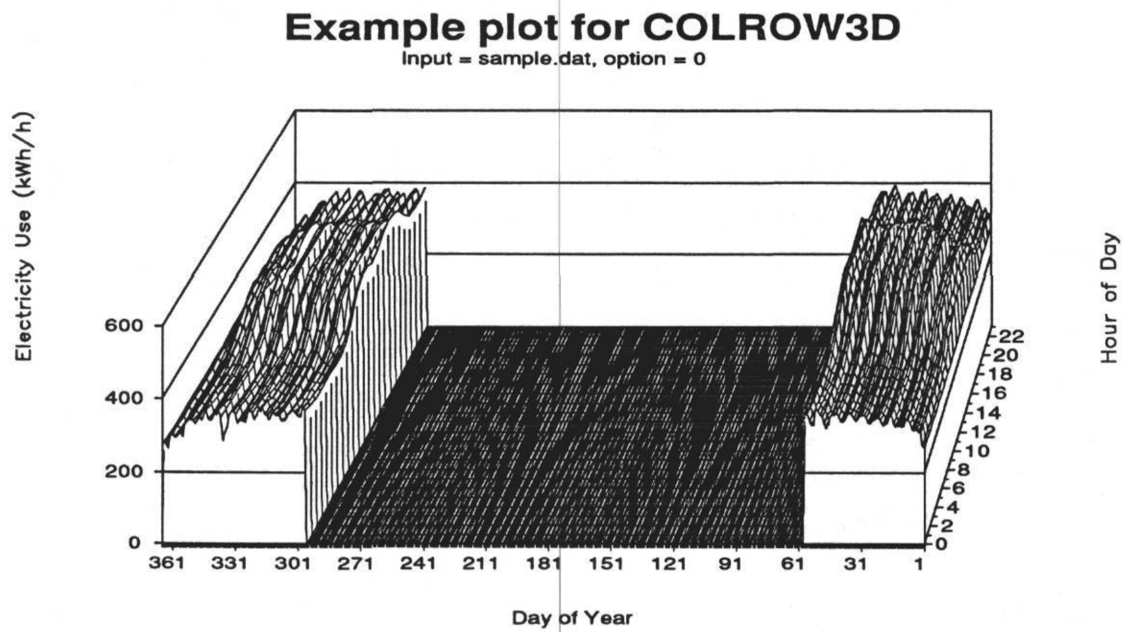
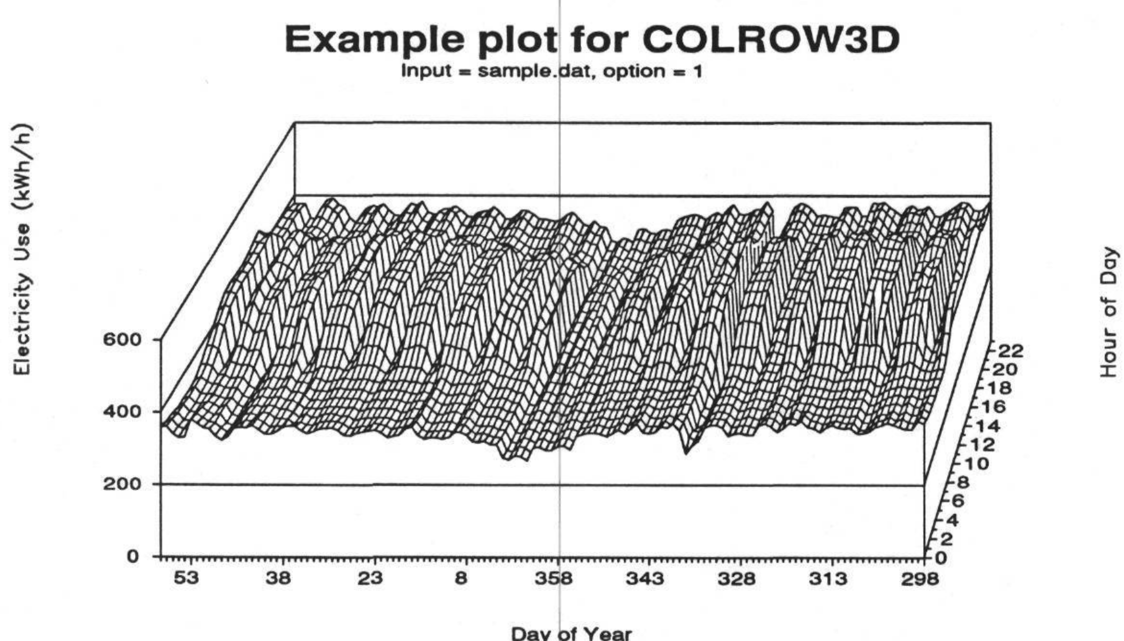


FIGURE 4-8 Example .PIC plots using the COLROW3D software package.





5.0 REFERENCES AND RELATED READING.

AIR 1992. AIR: Software for Calculating Psychrometric Properties, S. Katipamula, R. Sparks, J.Spadaro, Energy Systems Laboratory, Texas A&M University, (January).

Agar, M. 1980. The Professional Stranger: An Informal Introduction to Ethnography, Academic Press, New York, ISBN-0-12-043850.

Aho, A., Kernighan, B., Weinberger, P. 1988. The AWK Programming Language, Addison-Wesley Computer Science Series, Reading, MA, ISBN 0-201-07981-X.

ASHRAE 1987. Handbook of HVAC Systems and Applications, American Society of Heating, Refrigeration and Air-Conditioning Engineers, Atlanta, Georgia, Chapter 48 - Energy Management.

ASHRAE 1977 Brochure on Psychrometry, Prepared by the ASHRAE Technical Committee on Psychrometry (1969-1972), ISBN-0-910110-07-7.

ASHRAE 1989. Handbook of Fundamentals, American Society of Heating, Refrigeration and Air-Conditioning Engineers, Atlanta, Georgia, Chapter 13 - Measurement and Instrumentation.

ASHRAE 1990. Handbook: HVAC Applications, Chapter 37 -- Building Energy Monitoring, Atlanta, GA.

Baker,D., and Hurley, W. 1984. "On-Site Calibration of Flow Metering Systems Installed in Buildings," NBS Building Science Series Report No. 159, (January).

Bendat, J. S., and Piersol, A. G. 1986. Random Data Analysis and Measurement Procedures, John Wiley and Sons, New York, ISBN-0-471-04000-2.

Benedict, R. 1984. Fundamentals of Temperature, Pressure, and Flow Measurement. John Wiley and Sons, New York, N.Y., ISBN 0-471-89383-8.

Bevington, P. R., Robinson, D.K., 1992. Data Reduction and Error Analysis for the Physical Sciences, 2nd Edition, McGraw-Hill, New York, ISBN-0-07-911243-9.

Borland 1990, Paradox, Borland International, Scotts Valley, California.

Bronson, D., Hinchey, S., Haberl, J., O'Neal, D. 1992. "A Procedure for Calibrating the DOE-2 Simulation Program to Non-Weather-Dependent Measured Loads", ASHRAE Transactions, Vol. 98, pt. 1.

Bryant, J., O'Neal, D. 1992. "Calibration of Relative Humidity Transducers for use in the Texas LoanSTAR Program", Proceedings of the 1992 Hot and Humid Conference, Texas A&M University, Energy Systems Laboratory Report No. ESL-PA-92/02-15.

Clark, D. R. 1985. "HVACSIM+ Building Systems and Equipment Simulation Program", User's Guide and Reference Manual, National Bureau of Standards - Reports No. NBSIR-84-2996 and NBSIR-85-3243.

Claridge, D., Haberl, J., Katipamula, S., O'Neal, D., Ruch, D., Chen, L., Heneghan, T., Hinchey, S., Kissock, K., Wang, J. 1990. "Analysis of Texas LoanSTAR Data", Proceedings of the Seventh Annual Symposium on Improving Building Systems in Hot and Humid Climates, Texas A&M University, College Station, Texas, October.

COLROW3D 1991. COLROW3D: Software for creating 3-D plots, J. Matson, R. Sparks, J.Spadaro, J.Haberl, Energy Systems Laboratory, Texas A&M University, (October).

Cortina, V. (ed.) 1988. "Precision Humidity Analysis", EG&G Environmental Equipment, 151 Bear Hill Road, Waltham, MA, 02154, (IR sensors).

Date, C. 1989. A Guide to SQL/DS, Addison Wesley Publishing Co., Reading Massachusetts.

DeCicco, J. M., and Kempton, W. 1987. "Behavioral Determinants of Energy Consumption in a Centrally-Heated Apartment Building", Energy Systems and Policy, Vol. 11.

Doebelin, E. 1990. Measurement Systems. McGraw-Hill, New York, N.Y., ISBN 0-07-017338-9, ISBN 0-07-017338-9.

EIA 1989. Consumer Electronics Bus (CEBus) - Home Automation Standard, Electronic Industries Association, 2001 Eye Street, N.W., Washington, D.C., 20006.

EEI 1981. Handbook for Electricity Metering, Edison Electric Institute, Washington, D.C., ISBN-0-931032-11-3.

Englander, S., Reynolds, C., Haberl, J. 1990. "The Princeton Boiler Plant Electronic Logbook Project", Proceedings of the ACEEE 1990 Summer Study on Energy Efficiency in Buildings, ACEEE, Washington, D.C., August.

Fels, M. (ed.) 1986. "Special Issue Devoted to Measuring Energy Savings, The Princeton Scorekeeping Method (PRISM)", Energy and Buildings, Vol. 9, Nos. 1 and 2.

Feuermann, D., Kempton, W. 1987, "ARCHIVE: Software for the Management of Field Data", Center for Energy and Environmental Studies Report No.216, (This also includes Tony's Tools, and Art's Tools which are useful columnar data processing tools), Princeton University.

Fracastoro, G. V., and Lyberg, M. D. 1983. Guiding Principles Concerning Design of Experiments, Instrumentation and Measuring Techniques, International Energy Agency and the Swedish Council for Building Research, Stockholm, Sweden.

FSF 1989. GAWK, Free Software Foundation (PC version of the Unix-based AWK toolkit), 675 Massachusetts Ave., Cambridge, Massachusetts 02139.

Gilmore, V. E. 1988. "Smart House", Popular Science, August, (An article that describes the Smart House project being developed by the Smart House Limited Partnership, Washington, D.C.).

Golden 1990. GRAPHER and Surfer, Golden Software, 809 14th Street, P.O. Box 281, Golden, Colorado, 80402-0281.

Haberl, J., Bronson, D., O'Neal, D. 1992. "An Evaluation of the Impact of Using Measured Weather Data Versus TMY Weather Data in a DOE-2 Simulation of an Existing Building in Central Texas," ESL-TR-91/08-10, to be published in the ASHRAE Transactions, July.

Haberl, J., and Claridge, D. 1987. "An Expert System for Building Energy Consumption Analysis: Prototype Results", ASHRAE Transactions, Vol. 93, Part 1.

Haberl, J., Claridge, D., Harrje, D. 1990b. "The Design of Field Experiments and Demonstrations", Proceedings of the IEA Field Monitoring Workshop, Gothenburg, Sweden, April.

Haberl, J., Englander, S., Reynolds, C., Nyquist, T., McKay, M. 1989. "Whole Campus Performance Analysis Methods", Sixth Annual Symposium on Improving Building Systems in Hot and Humid Climates, Texas A&M University, College Station, Texas, October.

Haberl, J., Katipamula, S., Willis, D., Weber, K., Matson, J., Rayaprolu, M., Subramanian, U. 1990a. "The Texas LoanSTAR Program: Acquiring and Archiving LoanSTAR Data", Proceedings of the Seventh Annual Symposium on Improving Building Systems in Hot and Humid Climates, Texas A&M University, College Station, Texas, October.

Haberl, J., and Komor, P. 1989 "Investigating an Analytical Basis for Improving Commercial Building Energy Audits: Early Results from a New Jersey Mall", Proceedings of the Thermal Performance of the Exterior Envelopes of Buildings IV, ASHRAE/DOE/BTECC/CIBSE, Orlando, Florida, December.

Haberl, J., Norford, L., Spadaro, J. 1989. "Diagnosing Building Operational Problems: Intelligent Systems for Diagnosing Operational Problems in HVAC Systems", ASHRAE Journal, June.

Haberl, J., Turner, W.D., Finstad, C., Scott, F., Bryant, J. 1992. "Calibration of Flowmeters for use in HVAC Systems Monitoring", Proceedings of the 1992 ASME/JSES/KSES International Solar Energy Conference.

Haberl, J., and Vajda, J. 1988. "Use of Metered Data Analysis to Improve Building Operation and Maintenance: Early Results From Two Federal Complexes", Proceedings of the ACEEE 1988 Summer Study on Energy Efficiency in buildings, Volume 3.

Harding, J. (ed). 1982. "Recent Advances in Chilled Mirror Hygrometry", General Eastern Corporation Technical Bulletin, 50 Hunt St., Watertown, MA, 02172.

Hart, G. 1985. "Non-Intrusive Appliance Load Data Acquisition", Proceedings of the International Load Management Conference, Section 40, Sponsored by the Electric Power Research Institute - Report No. EPRI-EM4643, Palo Alto, Calif.

Harrje, D. T. 1982. "Monitoring Energy Use - What is Needed?", Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings, August.

Harrje, D. T., Kirkpatrick, D. L., Norford, L. K., and Seroussi, R. E. 1984. "Data Collection and Analysis Hardware for Measuring Building Energy Use", Doing Better Setting an Agenda for the Second Decade, ACEEE, Vol. C, pp. 177-192, August.

Harrje, D. T. 1986. "Obtaining Building Energy Data, Problems and Solutions", Proceedings of the Field Data Acquisition for Building and Equipment Energy-Use Monitoring Workshop, ORNL Publication No. CONF-8510218, pp. 205-209.

Haves, P., and Trewhella, D. 1988. "Towards an Environment for HVAC Control System Evaluation", Proceedings of the USER-1 Building Simulation Conference, Society for Computer Simulation, September.

Howell, R., and Sauer, H. 1985. Environmental Control Principles: A Textbook Supplement to the ASHRAE 1985 Handbook of Fundamentals, ASHRAE, Atlanta, GA, ISBN-0-910110-43-3.

Hsieh, E. H., Norford, L. K., Socolow, R. H. 1989. "Calibrated Computer Models to Track Building Energy Use: The Role of the Tenant and Operator Decisions.", Energy and Buildings, in preparation.

Huang, P. 1991. "Humidity Measurements and Calibration Standards", ASHRAE Transactions, Vol. 97, p. 3521.

Hyland, R. W., and Hurley, C. W. 1983. General Guidelines for the On-Site Calibration of Humidity and Moisture Control Systems in Buildings, N.B.S. Building Science Series 157, September.

Hurley, C. W., and Schooley, J. F. 1984. Calibration of Temperature Measurement Systems Installed in Buildings, N.B.S. Building Science Series 153, January.

Hurley, W. 1985. "Measurement of Temperature, Humidity, and Fluid Flow", Field Data Acquisition for Building and Equipment Energy-use Monitoring, ORNL Report No. CONF-8510218, (March).

Intex 1990. 3-D Graphics, Intex Solutions, 161 Highland Ave., Needham, Massachusetts 02194 (requires Lotus 123).

ISA 1976. Recommended Environments for Standards Laboratories, Instrument Society of America - Recommend Practice, Research Triangle Park, North Carolina.

Jilar, T. 1990. Proceedings of the Field Monitoring for a Purpose Workshop, Gothenburg, Sweden, Sponsored by the Swedish Council for Building Research, Stockholm Sweden, and prepared by the Building Services Engineering Department, Chalmers University of Technology

Katipamula, S., K., and Haberl, J. 1991. "A Methodology to Identify Diurnal Load Shapes for Non-Weather Dependent Electric End-Uses", Solar Engineering, 1991: Proceedings of the ASME-JSES -JSME International Solar Energy Conference, pp. 457-467, Reno, NV, March.

Klein, S. A., Beckman, W. A., and Duffie, J. A. 1976. "TRNSYS - A Transient Simulation Program", ASHRAE Transactions, Vol. 82, Pt. 2.

Komor, P., Kempton, W., and Haberl, J. 1989. "Energy Use, Information, and Behavior in Small Commercial Buildings," Center for Energy and Environmental Analysis - Report No. 240, Princeton University, July.

Kulwicki, B. 1991. "Humidity Sensors", Journal of the American Ceramic Society, Vol. 74, pp. 697-707.

Lee, I. (ed) 1988. "Review of Humidity Sensors", Texas Instruments Incorporated, Attleboro, MA, 02703 (general reference).

Leider, M. et al. 1990. "A Solid State Amperometric Humidity Sensor", Journal of Applied Electrochemistry, Chapman and Hill: Vol. 20, pp. 964-8.

Lotus 1985, Lotus 1-2-3 Spreadsheet, Lotus Development Corp., 55 Cambridge Parkway, Cambridge, Massachusetts 02142.

Lyberg, M. D. (ed.) 1987. Source Book For Energy Auditors, Volume 1 and 2, International Energy Agency and the Swedish Council for Building Research, Stockholm, Sweden.

MacDonald, J. M., Sharp, T. R., and Gettings, M. B. 1989. A Protocol for Monitoring Energy Efficiency Improvements in Commercial and Related Buildings, Oak Ridge National Laboratory Report ORNL/CON-291, September.

MacDonald, J. M. and Wasserman, D. M. 1989. "Investigation of Metered Data Analysis Methods for Commercial and Related Buildings", Oak Ridge National Laboratory Report No. ORNL/CON-279, May.

McQuiston, F., C., and Parker, J. D. 1988. Heating, Ventilating, and Air Conditioning. John Wiley and Sons, New York, ISBN 0-471-63757-2.

Miller, R. C. 1989. "Identification of Acoustic Signals Using Artificial Neural Networks", Master's Thesis, Marquette University, Milwaukee, Wisconsin, December.

Miller, R. 1989. Flow Measurement Handbook, McGraw Hill Publishing Company, New York, N.Y., ISBN 0-07-042046-7.

Morrissey, C. J. 1990. "Acoustic Humidity Sensor", NASA Tech Brief. Vol. 14, #19, April, (acoustic).

Norford, L., Mabey, N. 1992. "Non-intrusive Electric Load Monitoring in Commercial Buildingsdsd", Proceedings of the Eighth Symposium on Improving Building Systems in Hot and Humid Climates, published by the Energy Systems Laboratory, Texas A&M University.

Norford, L., Tabors, R., Byrd, G. 1992. "Non-Intrusive Electrical Load Monitoring, a Technique for Reduced-Cost Load Research and Energy Management: Results of Field Tests and Recent Developments", Proceedings of the 1992 ACEEE Summer Study, Asilomar, CA.

Nutter, D., Britton, A., Muraya, N., Heffington, W. 1990. "LoanSTAR Energy Conservation Audits: January 1989 - August 1990", Proceedings of the Seventh Annual Symposium on Improving Building Systems in Hot and Humid Climates, Texas A&M University, College Station, Texas, October.

O'Neal, D., Bryant, J., Turner, W., Glass, M. 1990. "Metering and Calibration in LoanSTAR Buildings", Proceedings of the Seventh Annual Symposium on Improving Building Systems in Hot and Humid Climates, Texas A&M University, College Station, Texas, October.

Olsen, K., Komor, P., and Fels, M. 1988. "Development of a Data Base to Characterize the Small Commercial Sector: A New Jersey Example", Proceedings of the ACEEE 1988 Summer Study on Energy Efficiency in buildings, Volume 10.

Olsen, L. 1974. Introduction to Liquid Flow Metering and Calibration of Liquid Flowmeters, N.B.S. Technical Note 831.

Rabl, A. 1988. "Parameter Estimation in Buildings: Methods for Dynamic Analysis of Measured Energy Use", ASME Journal of Solar Energy Engineering, Vol. 110.

Ramboz, J. D. and McAuliff, R. C. 1983. A Calibration Service for Wattmeters and Watt-hour Meters, N.B.S. Technical Note 1179.

Reddy, T. A. 1989. "Application of Dynamic Building Inverse Models to Three Occupied Residences Monitored Non-Intrusively", Proceedings of the Thermal Performance of the Exterior Envelopes of Buildings IV, ASHRAE/DOE/BTECC/CIBSE, Orlando, Florida, December.

Robinson, J., Bryant, J., Haberl, J, Turner, D. 1992. "Calibration of Tangential Paddlewheel Insertion Flowmeters", Proceedings of the 1992 Hot and Humid Conference, Texas A&M University, Energy Systems Laboratory Report No. ESL-PA-92/02-09.

Ross, I., J., and White, G. M. 1990. "Humidity", Instrumentation and Measurement for Environmental Sciences: Transactions of the ASAE, 2nd ed., p. 8-01.

Ruch, D., Chen, L, Haberl, J., Claridge, D. 1991. "A Change-Point Principal Component Analysis (CP/PCA) Method for Predicting Energy Usage in Commercial Buildings: The PCA Model", Solar Engineering 1991: Proceedings of the ASME-JSES-JSME International Solar Energy Conference, pp.441-448, Reno, Nevada, March.

Ruch, D., Claridge, D. 1991. "A Four Parameter Change-Point Model for Predicting Energy Consumption in Commercial Buildings", Solar Engineering 1991: Proceedings of the ASME-JSES -JSME International Solar Energy Conference, Reno, Nevada, March.

SAS 1990, Statistical Analysis Software, SAS Institute, SAS Circle, Box 8000, Cary, North Carolina.

Schuster, G. 1985. "Field Data Acquisition Hardware", Field Data Acquisition for Building and Equipment Energy-use Monitoring, ORNL Report No. CONF-8510218, (March).

Sonderegger, R. C. 1978b. "Diagnostic Tests Determining the Thermal Response of a House." ASHRAE Transactions, Vol. 84, Part 1, pp. 691-702.

Subbarao, K. 1988. "PSTAR- Primary and Secondary Terms Analysis and Renormalization: A Unified Approach to Building Energy Simulations and Short-term Monitoring", Solar Energy Research Institute - Report No. SERI/TR-254-3175.

Synergistic. 1990. Software, Installation, and Technical Specifications for the Model C180 Survey Meter/Recorder, 5725 Bundy Rd., New Orleans, LA 70127.

Taylor, J. 1981. An Introduction to Error Analysis, University Science Books, Oxford University Press, Mill Valley, CA, ISBN 0-935702-07-5.

Sonderegger, R. C., 1978a. "Movers and Stayers: The Resident's Contribution to Variation Across Houses in Energy Consumption for Space Heating", in Saving Energy in the Home, Robert Socolow (ed.), Ballinger Publishing Co.

Ternes, M. P. 1987. Single-Family Building Retrofit Performance Monitoring Protocol: Data Specification Guideline, Oak Ridge National Laboratory Report ORNL/CON-196.

TeX 1986. The TEXbook, Donald Knuth, The American Mathematical Society and the Addison Wesley Publishing Company, Reading, Massachusetts.

Threlkeld, J. 1970. Thermal Environmental Engineering, Prentice-Hall, Englewood Cliffs, N.J., (general reference).

Treado S. J., and Bean, J. W. 1988. The Interaction of Lighting, Heating and Cooling Systems in Buildings - Interim Report, National Institute of Standards and Technology, NISTIR 88-3860, December.

Treado, S. J., 1988. Experimental Plan for Investigation of Lighting and HVAC Interactions, National Institute of Standards and Technology, NISTIR 87-3656, February.

Turner, W., D. 1990. "Overview of the Texas LoanSTAR Monitoring Program", Proceedings of the Seventh Annual Symposium on Improving Building Systems in Hot and Humid Climates, Texas A&M University, College Station, Texas, October.

Turner, W., D., Haberl, J., Bryant, J., Finstad, C., Robinson, J. 1992. "Calibration Facility for the LoanSTAR Program", Proceedings of the 1992 ASME/JSES/KSES International Solar Energy Conference.

Tufte, E. R. 1983. The Visual Display of Quantitative Information, Graphics Press, Cheshire, Connecticut.

Ullman, J. 1988. Principles of Database and Knowledge-base systems, Computer Science Press, Rockville, Maryland.

Verdict, M., Haberl, J., Claridge, D., O'Neal, D., Heffington, W., Turner, D. 1990. "Monitoring $98 Million in Energy Efficiency Retrofits: The Texas LoanSTAR Program", Proceedings of the ACEEE 1990 Summer Study on Energy Efficiency in Buildings, ACEEE, Washington, D.C., August.

Wiesman, S. (ed.) 1989. "Measuring Humidity in Test Chambers", General Eastern Corporation, 50 Hunt Street, Watertown, MA 02172. (chilled mirror dewpoint sensors).

Wise, J. A., Soulen, R. J. 1986. Thermometer Calibration: A Model for State Calibration Laboratories, N.B.S. Monograph 174, January.

Wise, J. A. 1976. Liquid-In-Glass Thermometry, N.B.S. Monograph 150, January.

Zuboff, S. 1988. In the Age of the Smart Machine: The Future of Work and Power, Basic Books, New York, N.Y.

6. ACKNOWLEDGMENTS

This work was funded through the Texas Governor's Energy Office as part of the LoanSTAR program. I would like to thank my fellow researchers at the ESL for helpful comments in preparing this workbook. Special thanks are also in order for Ms. Daralyn Wallace and Ms. Mary Castillo who put up with us through numerous editing sessions.

The idea and motivation for this workbook and the accompanying software belong to Dr. Steve Jaeger, at the Texas Governor's Energy Office, who felt that the LoanSTAR monitoring and analysis procedures and routines were too important to be kept only within the program.

7. APPENDIX

"Field Data Acquisition Hardware", G. J. Schuster, Field Data Acquisition for Building and Equipment Energy-use Monitoring, ORNL Report No. CONF-8510218, (March 1985).

"ARCHIVE: Software for Management of Field Data", D.Feuermann, W. Kempton, Center for Energy and Environmental Studies Report No. 216, Princeton University, June 1987.

Field Data Acquisition Hardware

George J. Schuster
Engineering Physics Department
Pacific Northwest Laboratory
Richland, Washington

1.0 Introduction

Direct metering of electrical energy flows to specific end-uses in commercial buildings is an important adjunct to evaluation of possible conservation savings as a result of weatherization or equipment optimization measures. In addition, metering can often represent the most reliable and low-cost means of determining the weekly operating schedules required to model the building computer simulation tools. In order to support a number of building studies sponsored by the U.S. Department of Energy (DOE) and later by the Bonneville Power Administration, an effort was initiated at Pacific Northwest Laboratory (PNL) to develop a low-cost method for carrying out such metering.

After several pilot studies, a decision was made to engineer a microprocessor-based field data acquisition system (FDAS) tailored for the specific job of metering electrical energy flows in complex commercial buildings. The applications indicated that the field unit should have the following characteristics:

- It should permit the metering of as many circuits as necessary at a given site to provide the desired level of end-use disaggregation and redundancy in measurement.
- It should permit the acquisition of data at a variety of temporal resolutions, ranging from intervals short enough to monitor the behavior of electrical equipment with short cycle times to hourly or lower resolution data collection. If possible, it should permit remote adjustment of the measurement resolution.
- It should be reliable, maintaining its performance over extended periods in the field without requiring adjustment.
- It should be sufficiently inexpensive to permit metering of a large sample of buildings at reasonable cost.
- It should have nonvolatile, solid state memory and permit the remote acquisition of data.
- It should permit measurements with an accuracy at least equal to that of utility-grade meters.
- It should be capable of supporting a variety of instrumentation, including not only watt metering sensors but also meteorological devices.

The most recently completed system, designed to meet these requirements, is currently being deployed in BPA's End-Use Load and Conservation Assessment Program (ELCAP), an end-use metering study of 1000 residential and commercial buildings in BPA's service territory. This document provides a functional description of the FDAS design resulting from this support. A section on the hardware describes in some detail the functionality of the FDAS, and is followed by a description of the sensors which measure energy use and other parameters that affect energy use. The final section provides a description of the system's software.

2.0 FDAS Hardware

The principle components of the FDAS are the data logger, the modem, and the watt meter circuits. The system also includes a dc power supply, sampling transformers, fuses, and the ac input terminal blocks. These components are mounted to a single anodized aluminum panel for easy assembly and testing and for quick removal and replacement during installation and repair. All components are then placed in a standard electrical enclosure, and the modem board is mounted to the door (Figure 1).

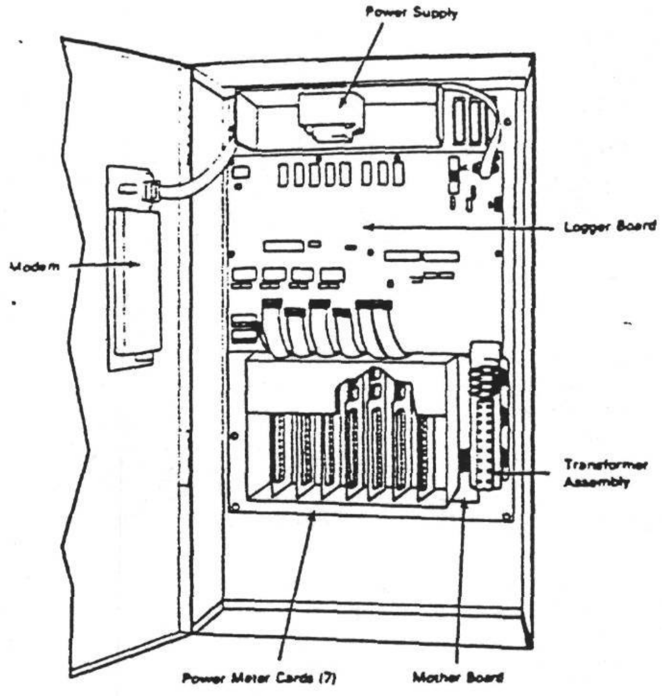


Figure 1

Layout of Digital Field Data Acquisition System

All system interconnects are accomplished with standard industry insulation displacement connectors and ribbon cable or, in the case of power cables, 18-gauge (UL-listed) stranded interconnect wire. The unit was carefully designed to route all high-voltage wire away from low-voltage sections, and wire harnesses were securely fastened using standard industrial techniques.

.1 Data Logger

The FDAS data logger contains the microcomputer or central processing unit (CPU), which has the software needed to run the FDAS, as well as interfaces to each watt meter board. The data logger also provides a standard serial link to the modem, which enables the FDAS to use an ordinary telephone line as a means of receiving commands from and transferring data to a central data acquisition computer. A block diagram of the logger is given in Figure 2.

The data logger runs a program (described in Section 4) that acquires data periodically from each watt meter board and stores these data in nonvolatile memory. Sixteen kilobytes of data storage are available in this memory section where data are accumulated for each input channel. Data for each channel are collected by converting the watt meter's analog signal into a digital number using an analog-to-digital (A/D) converter; these converters are compatible with the 8-bit microprocessor bus and have 16 multiplexed inputs each. Thus, each data logger has the capability of using four A/D converters, with 16 channels each, for a total of 64 analog input channels. A few of these analog input channels are usually provided with additional circuitry to enable the data logger to collect meteorology data, including wind direction, inside and outside air temperature, horizontal solar radiation, and humidity. The CPU selects each of the 16 channels of each A/D converter and converts the 0-5 V dc signal at its input to a single 8-bit number. This number is then stored in memory and accumulated into totals for data reduction and storage. Providing the A/D converters with an analog reference voltage is a precision reference that can be preset for accuracy at the time the FDAS is calibrated.

The CPU is a single-chip microcomputer that has 128 bytes of Random Access Memory (RAM), 2048 bytes of Erasable Programmable Read Only Memory (EPROM), 29 parallel input/output lines, a serial communications interface, and a 16-bit programmable timer.

The 16,384 (16K) bytes of memory are implemented with eight-2K Complementary Metal Oxide Semiconductor (CMOS) RAM chips that are configured in a battery-backed protection circuit. If power is lost to the FDAS, the battery-protected memory retains data through the power outage. In addition to this memory protection function, the CPU is programmed to dial out via the modem to the central computer, signaling power outage and providing its identification number. The FDAS resumes logging data after the power outage, and the time is reset by the central computer shortly after the condition is detected. A special reset circuit was designed that detects brownout conditions in advance of power failure and holds the CPU in a reset condition to prevent random accessing of program or data storage. Once power is restored, the CPU is released to execute its program under normal conditions.

Three independent digital input devices, or Parallel Interface Adapters (PIAs), give the FDAS an additional 48 channels of digital input capability. These PIAs allow external monitoring or control of devices that use a standard Transistor-Transistor Logic (TTL) or 5-V interface. The CPU is programmed to count pulses at the PIA inputs at rates up to 75 cycles per second; this enables the FDAS to accept input from devices such as pulse-initiating watt-hour meters. This additional parallel capability enables the FDAS to be coinstrumented with other energy-monitoring equipment or to act as the remote data logging unit for studies that use these devices. Two of these digital inputs have conditioning circuits to convert sine waves generated by wind speed anemometers to digital inputs.

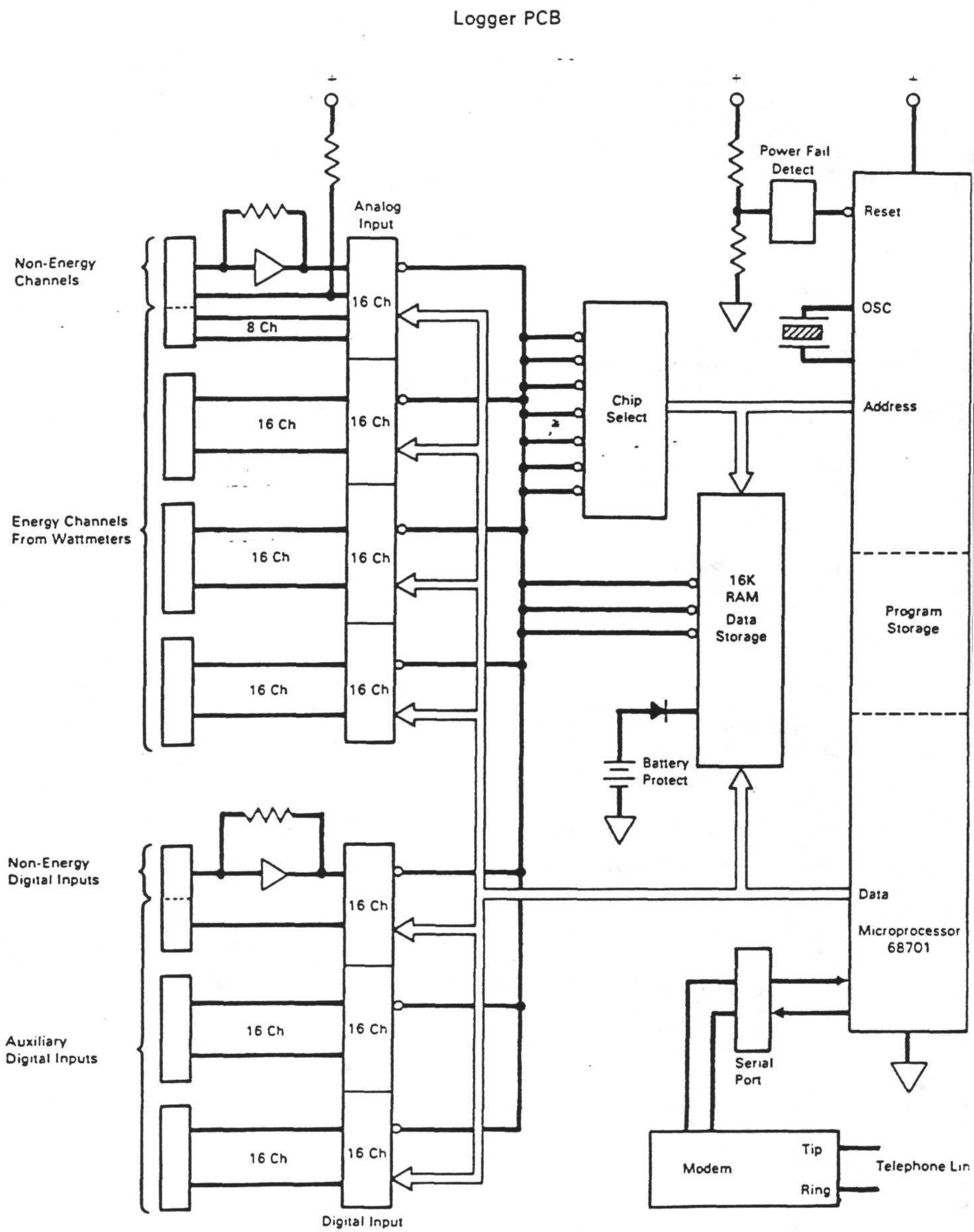


Figure 2

Logger Block Diagram

3.0 Sensors

The sensors used with the FDAS can be ouped into two categories: 1) sensors that llow the logger to monitor electrical energy onsumption, and 2) meteorology sensors, which low the logger to monitor important param- ers that may affect energy consumption in a uilding.

1 A Sampling Watt Meter Circuit Board

Any electrical power meter must accumu- ate instantaneous products of voltage and rrent. In the sampling watt meter reported re, voltages are brought into each watt eter using line voltage stepdown trans- rmers. These sampling transformers are tached to each voltage source and provide e reference voltages that are associated ith the phases found in an electrical distri- tion panel. Current sensing is provided by stalling a current transformer (CT) around each breaker wire to be monitored. These devices, which resemble a small plastic dough- t with two lead wires, are used by passing e wire to be sensed through the hole and attaching the two lead wires to the appropri- ate channel input on the watt meter board. As rrent flows from the circuit breaker to the pliance or other load, the CT develops a voltage between the two leads proportional to he load current.

The circuit in Figure 3 uses the trans- former assembly to develop three reference ltages which are converted to two pulse tputs, one for the positive half-cycle of the waveform and one for the negative half cycle. These six pulse outputs, two for each eference voltage, are then picked up by the att meter cards, which have the CTs outputs applied to them. As the amplitude varies from the voltage sensing transformer, the pulse tputs respond proportionally and the result- g reference voltages contain the waveform information in a digital form.

Scaling resistors are used to accommodate e use of the transformer assembly on cir- cuits from 120 V up to 480 V. These scaling esistors must be selected precisely to pro- ide the digital circuitry with voltages ithin its capability and directly propor- tional to voltages associated with the instal- ation. The voltage amplitudes and phase elationships are maintained so as to provide real-time multiplication of voltage and cur- rent within the FDAS.

Switches are provided on the watt meter card to select the pair of pulses from the mother board for each CT; therefore, each hannel can be configured for any of the three hases. A scaling resistor is provided to adjust the full-scale range of the CTs size as it does on the analog FDAS. The CT input, once adjusted for full scale, is gated by the reference pulses into an integration amplifier whose output is ribbon-cabled to the logger board. Since these reference pulses are pro- portional to phase angle and voltage ampli- tude, this digital method of watt metering provides accurate power measurements for which the power factor of a circuit is already accounted.

3.2 Full Sensor Complement

In addition to the watt meters, the FDAS can be equipped with an indoor temperature sensor and with meteorological sensors to measure wind speed, wind direction, solar radiation, and outdoor air temperature. Sites with meteorological stations would also receive an indoor relative humidity sensor. The meteorological instruments would be mounted on top of a pole of sufficient height to ensure that the effects of the building's envelope do not interfere with the instru- ment's performance. Signals from the instru- ments would be fed to special signal condi- tioning circuits within the FDAS.

Indoor air temperature is measured by using a calibrated reference voltage across a resistor divider network containing a temperature-sensitive thermistor. This thermistor (a Yellow Springs YSI 44006) provides an approximately linear resistance change versus temperature in the 0° to 30°C range. From this change the resistor divider network provides a temperature-dependent voltage which is input into an analog chan- nel. Despite the slight nonlinearity in the resistance temperature characteristic curve and uncertainties due to a digitization scheme, the combination provides temperature measurements with an absolute accuracy of ±0.5°C over this range.

Wind speed is measured with a 3-cup drag- type anemometer (Weather Measure W200-SD) attached to the shaft of an ac generator. The anemometer has a usable range of 0 to 100 mph with a 5% accuracy and a 1.33 mph resolution with the A/D converter system. The ac signal from this instrument is rectified into a square wave with a frequency proportional to wind speed. This digital signal is input to one of the digital channels on the data logger.

The wind direction vane is mounted on a common axis with the anemometer and supported on teflon thrust bearings. A wiper contacts a potentiometer to create a voltage divider net- work. This device has a 0 to 5 V (dc) output for a rotation of 0° to 360° clockwise from north. The resolution of the instrument com- bined with a data collection system is 1.4°.

Digital Power Meter

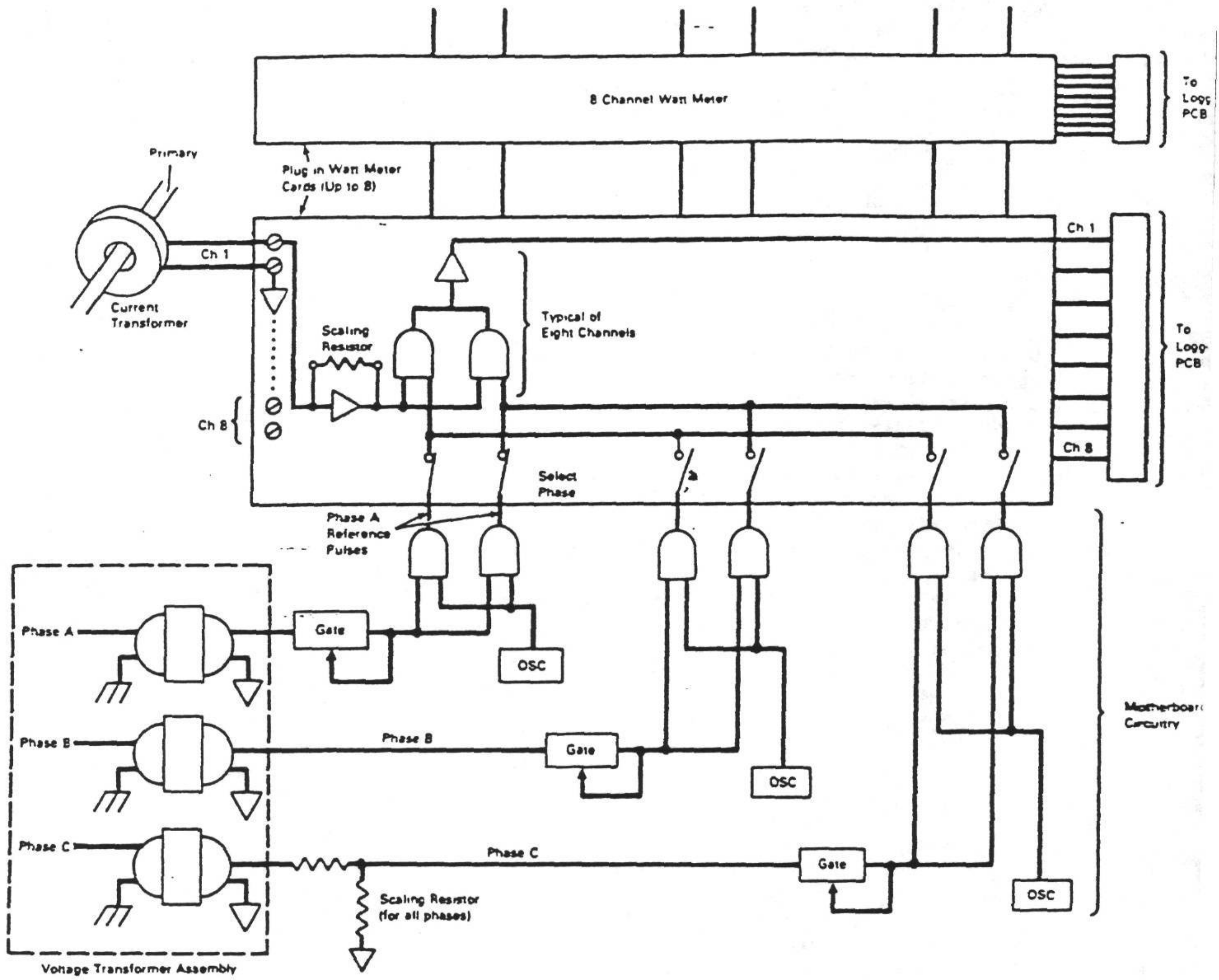


Figure 3

Digital Watt Meter

The accuracy also depends on an accurate determination of true north at the time of installation.

A pyranometer (Li-Cor LI-200S) is used to measure the solar radiation that strikes a horizontal surface. A silicon photodiode is used as the sensing element with an output of near 80 μ amps at full sun (1 kW/m^2). Each instrument has its own characteristics and thus its own calibration constant. Typical resolution is 5 W/m^2 with an accuracy of ±5% for incident angles of less than 80°.

The outdoor air temperature sensor uses a special thermistor (Yellow Springs YSE 44211A) in an active circuit to produce a linear response over a wide temperature range. The accuracy of the instrument is ±1.0°C over a range of -50° to 60°C. The outdoor air temperature sensor is mounted on the same pole as the meteorological station in a shroud that protects it from direct solar radiation.

The indoor relative humidity sensor (Weather Measure Model 5131-A) is a variable capacitance instrument that provides 0 to 5 V (dc) output proportional to 0 to 100% relative humidity. This instrument, like the indoor temperature sensor, is mounted on an interior wall and the resolution of the instrument is 0.4% relative humidity with an accuracy of ±5%.

One additional sensor is used to monitor woodstoves used as space heating equipment. A thermocouple (Omega XC1B-111) placed in the chimney transmits a low voltage signal proportional to the hot flue gas temperature to the conditioning board when the woodstove is in use. This signal is amplified and recorded by one of the data logger's nonenergy channels.

The full complement of exterior sensors available for use in the FDAS allows all the significant parameters of building energy use to be monitored, making the FDAS a flexible and highly useful tool.

4.0 FDAS Software

The FDAS software processes the signals from the sensors into average values and reports those average values on request; that is, it processes signals from a collection of sensors into values that are representative of a time interval. This software must perform the repetitive scans and data manipulations necessary for the measurements to constitute a well-defined, appropriate physical quantity and to allow the analyst to examine the performance of his apparatus and retrieve the measurements upon request without disturbing the data collection function. Note that the FDAS software does not reduce the data into its final form. The task of applying calibration constants to the data is left to the central data acquisition computer.

The data accumulation that occurs within the integration period is the principle software routine which is performed. The integration period is the interval of time over which the signals from the sensors are averaged; it is selected by the user and can be set to any integer number of seconds from 1 minute to 18 hours. This integration period determines the length of contiguous, nonoverlapping intervals of time.

The values from the scans from all 112 data channels are accumulated for each integration period. Each channel has three 8-bit bytes of memory; at the start of the integration period, their sums are all zero. The values from the sensors are added to these sums, and, at the end of the integration period, the sums are converted by the data reduction routine to a record of the integration period.

The data reduction routine uses data compression techniques to reduce the data from the 3-byte sums so that only significant numbers are saved in the time series records. The routine will save two bytes from sums requiring the highest significance, one byte where only 8 bits are significant, and nothing where the sum has no significance (no sensor).

In order to assess the performance of the FDAS and allow for the extraction of the data from its memory, the FDAS recognizes 19 commands, summarized in Table 1. The FDAS examines the incoming characters from the modem, if the character stream matches one of the 19 executable commands, then any old command is terminated and the new command is requested.

After the interpretation of a command, the command processor executes one of the routines for the control of the FDAS. This processing is assigned a lower priority than the interpretation of the commands, so the execution of a command can always be interrupted. A new command will always supersede the previous one. This is most useful for ending the execution of commands that initiate continuous displays.

Whenever the software is idle, the logger indicates the idle condition by lighting a small light-emitting diode (LED). Here the FDAS awaits the next timer interrupt. These interrupts indicate the need for additional data processing.

4.1 Data Records

The data records consist of the time-series data as collected by the FDAS, i.e., in binary format, directly from the sensors. The records consist of a header, the digital data, and the analog data.

Table 1

List of Available Logger Commands

Function
Continuously display pulse counts from the 48 digital input channels
Continuously display scans of the 64 analog input channels
Continuously display the 3-byte sums
Display the software identification
Display the control parameters
Enter the set mode; display pointer
Enter the point mode
Add 1 to pointer or parameter
Subtract 1 from pointer or parameter
Add 10 to pointer or parameter
Add 50 to pointer or parameter
Transmit the data records in ASCII
Transmit the first data record in binary
Transmit the next data record in binary
Retransmit a data record in binary
Reset software
Set control parameters to zero
Clear data records and tests memory
Reset the modem

The data records were designed for the efficient use of memory and for speed of transmission. The record is transmitted in the same form in which it is stored in memory, as an integer number of 8-bit binary numbers. The length of the record is determined by the configuration of the specific installation and the number of sensors connected to the FDAS. One or two bytes of data are entered into the data record for every active channel.

Each record will have a header of nine bytes for the purpose of identifying the record. The header was designed to verify the performance of the data acquisition system as well as to identify the data record. For these purposes, the data record provides a check sum, a record length check, and a time stamp. The check sum is the result of adding together the values from the other bytes in the data record. The record length check verifies the total number of bytes in the data record, and the time stamp ensures that all necessary data records have been reviewed and are in the proper order.

It is important to assess the validity of the data record as it is recovered. If the check sum, the record length, and the time stamp all agree with the values expected for these parameters, then the measurements contained in the record can be taken as an accurate representation of the scans of the channels. Other considerations are used to determine if the signals from the sensors are representative of the quantities being sensed.

Princeton University

School of Engineering/Applied Science
Center for Energy and Environmental Studies
The Engineering Quadrangle—Olden Street
P.O. Box CN5263
Princeton, New Jersey 08544-5263
Phone (609) 258-5445 FAX (609) 258-3661

13 August 1991

Prof. Jeff Haberl
Mechanical Engineering
Texas A&M University
College Station, TX 77843-3123

Dear Jeff,

You have requested permission to distribute the ARCHIVE program (for buildings data) which I designed and Daniel Feuermann and I developed. You propose to distribute the source code and documentation, with some modifications made by your team at A&M.

The software source code bears the following notations:

"Copyright (c) 1987, Princeton University."

"This program bears a copyright notice to prevent rights from being claimed by any other party. Princeton University intends that the program be placed in the public domain and grants permission for it to be used and redistributed, provided that: the source code is distributed, this notice is retained, and the program is not sold for profit."

"The program is distributed "as is". Princeton provides no warranty or support service."

The distribution you propose would be consistent with these terms. Please insure that the modifications and bug fixes you have made to the software are tested in production prior to distribution, and that such modifications are described in accompanying documentation.

We would prefer that you provide full printed documentation with the software, rather than on-disk documentation, but leave that decision to you. You have permission to photocopy our printed documentation, retaining the Princeton identification, possibly with supplementary sheets added under A&M's name. Charging for media and distribution costs is acceptable, and that is our practice here.

We are pleased that our software has furthered your efforts, and welcome your further publicizing it and redistributing it.

Sincerely,

Willett Kempton

Willett Kempton,
Research Anthropologist

xc: Rob Socolow

ARCHIVE: Software for

Management of Field Data

PU/CEES Report No. 216

June 1987

Daniel Feuermann and Willett Kempton

the center for energy and environmental studies

princeton university

ABSTRACT

Archive is a computer program which pre-processes field data recorded at regular intervals through time. It serves four functions:

- It cleans raw data, filtering data which are improperly formatted, outside plausible bounds, or garbled by transmission errors.
- It converts the raw data, performing any necessary algebraic transformations and formatting to produce a clean archive file for subsequent analysis.
- It ensures the keeping of a record of changes to the field instrumentation through time. The same record identifies the data in the archived file.
- It produces a separate log -- a file that identifies time and location of erroneous data in the raw data file.

Archive is designed to be used for automated data processing, and is therefore batch-oriented rather than interactive. It deals correctly with various calendrical systems, missing or erroneous data, and equipment changes through time. Archive was developed by a group of scientists collecting energy data on buildings. While it applies to any instrumentation data, it may best solve the problems faced in buildings instrumentation, such as malfunctioning or changing of sensors partway through a monitoring period, cumulative meters which wrap around to zero, and short-term experiments conducted in the middle of a long-term monitoring project. These and similar problems have in the past usually been solved by manual adjustments of the data or ad-hoc programming. Archive handles them uniformly and automatically, and leaves a record of what it has done. The program is in the public domain. It is written in Standard Pascal and runs on a variety of computer systems.

TABLE OF CONTENTS

I. INTRODUCTION

I.1 Why Did We Write Archive?

Archive was written to simultaneously solve several problems which arose in data collection and analysis projects at Princeton University's New Jersey Energy Conservation Lab. Archive makes range checking, data conversion, and format standardization easy to do, and properly formats the data for downstream analysis. While it is possible to perform checking, converting, and reformatting without a program like Archive, such manipulations are typically buried inside complex programs. Data checking in particiular poses the institutional problem that the criteria can not be easily reviewed, or even that routine checking is foregone because of the inconvenience of programming it. With Archive, data checks are more likely to be done from the beginning and the criteria used are easily readable by all levels of project staff.

In an active research institution with many simultaneous projects, Archive offers the additional advantage that data files are written in a standard format for archival and exchange of data and programming tools. The standardization of data format means that tools developed for one project can often be used unchanged on other projects. After just a year's use of Archive at Princeton, this tool sharing is already happening across projects using very different field instruments, which previously had produced incompatible data. (The projects range from measuring home radon concentrations, to air conditioning in multi-family buildings, to large commercial HVAC). The diverse raw data from these projects had previously required developing independent software "front-ends" for each project. All are now handled with a single tool - Archive.

I.2 Overview of Archive's operation

The Archive program has two input files and two output files. The primary files are the "raw data file", which it reads, and the "archived data file", which it creates. The other two files are the "channel table file", needed to identify variables in the raw data file, and the "log file" to which the program writes messages and error statistics.

The raw data file contains data as received from the field--possibly with errors, formatting problems, etc.--and the archived file contains the clean data to be used as the permanent archive for subsequent analysis. The archive file is clean in two senses: the data are formatted correctly and a standard missing value replaces any unreadable or implausible data values.

Archive is "crash-proof" to errors in the raw data file -- no matter how badly garbled the input file is, archive produces error reports and puts missing data values in the archive file rather than terminating with a program error. (However, program termination may be caused by errors in the channel table or by specifying files which do not exist.)

The channel table is central to the design of Archive. The researcher creates it, listing each of the data collection "channels" (i.e., variables or columns) recorded by the field instruments. For each channel, the channel table specifies how the raw data are to be transformed into archived data, including appropriate conversion formulas, checking procedures, and the format and position of the data in the archive file. In addition, each entry in the channel table is associated with a date and time which indicate the beginning date and time when the information about the channel becomes valid. (Of course, for the user to use this capability the field instrument must write some date, time or sequence number to the raw data file.) Therefore, channel specifications may be changed during different phases of the project. The new specifications are simply added to the end of the channel table file, with the times at which they become effective. Whenever Archive is run, it synchronizes the dates and times in the data file with those in the channel table. Thus, for example, several sequential raw data files can be processed by the same channel table, or an old raw data file can be re-archived, even with an updated channel table. In either case, the correct conversions and calibrations will automatically be applied.

The fourth file used by Archive is a log file. This is created every time Archive is run, and it lists errors and summary statistics for that run. In some projects, the log file will be used as a guide to fix parts of the raw data file using a text editor. Archive can then be re-run to produce a more complete archive file. Depending on the record-keeping policy of the individual project, the log file may be printed out and filed for each run, or it may be used only when there are errors needing attention.

A diagram of the files and data flow associated with Archive is given below:

```
                                                                  Analysis
                ________                          ________         programs:
               | raw data |                      | archived |
Field          |   file   |-------->|  _______  |------->|   data   |-->graphics
instru-        |          |         | ARCHIVE |        __|_  file   |-->statistics
ments -->|     |    | Channel |---->|_________|--->| log |          |   etc.
               |    |  table  |                  | file|          |
               |____|         |                  |     |__________|
                    |         |                  |     |
written by   -->|   |         |                  |     |--> inspect and file
investigator        |_________|                  |_____|    or discard
```

As illustrated by the diagram, Archive is a pre-processor or "filter", it does not replace analysis programs.

The figure on the following two pages gives an example of a raw data file and corresponding channel table file, along with the log and archive files they produce. The details shown in this figure have not yet been explained, and only one of many possible data formats are shown. It is shown now to provide an overview, since it gives a complete mapping across all the pieces which can be referred back to as each part is explained in more detail.

The following two pages
give an example data file
processed by Archive,
along with its channel table file
and output.

RAW DATA FILE

```
1  85001  100  10
2  1.4  21.2  10.0  1200.0
1  85001  200  12
2  1.6  21.5   9.8  1244.0
warning temperature T2 high
1  85001  300  10
2  1.7  216  10.1  1252.0
1  85001  400  10
2  1.6  21.8  10.5  1280.0
1  85001  500  10
2  4.3  22.1  10.8  1420.0
1  85001  600  10
2  4.0  22.3  10.9  1440.0
1  85001  700  10
2  A4.2 22.4  10.7  1462.0
1  85001  800 10
2  4.3  22.5  10.8  1480.0
1  85001  900  10
2  4.8  22.0  10.5  1506.0
```

CHANNEL TABLE FILE

Date MM/DD/YY	Time HH:mm	Raw line	Raw coln	Arch pos	Name of channel	Archive Units	Arch Format	Conv'n Code	Calib. or Constants	Error Code	Limitation Constants	Channel Description
#24												
01/01/85	01:00	1	1	1	JulDate	YYDDD	I5	20		0		
01/01/85	01:00	1	2	2	Time	HHmm	I5	15		0		
01/01/85	01:00	1	3	3	#polled	-	I4	1		0		# sampling
01/01/85	01:00	2	1	4	T1	deg C	F6.2	1		0		temperature
01/01/85	01:00	2	2	5	T2	deg C	F6.2	2	0.99 -0.5	1	0 200	temperature
01/01/85	01:00	2	3	6	flow	liter/s	F8.3	1		0		circulation
01/01/85	01:00	2	3	7	H2O vol	liter	F8.3	6	0.0012	1	0 2000	water use
01/01/85	01:00	2	4	8	gas	ft3	F7.2	1		0		gas meter
01/01/85	01:00	2	4	9	gas	ft3/s	F7.2	8	10 20000	1	0 200	gas usage
01/01/85	01:00	1	1	10	—	hours	F5.2	18		0	(virtual: time-gap)	
01/01/85	01:00	1	1	11	—	days	F9.4	28		0	(virtual: "decimal time")	

ARCHIVE FILE

```
85001   100   10    1.40  20.49    10.000  -99.000 1200.00   -99.00   0.00 1827.0417
85001   200   12    1.60  20.78     9.800   42.336 1244.00     0.12   1.00 1827.0833
85001   300   10    1.70 -99.00    10.100   43.632 1252.00     0.02   1.00 1827.1250
85001   400   10    1.60  21.08    10.500   45.360 1280.00     0.08   1.00 1827.1667
85001   500   10    4.30  21.38    10.800   46.656 1420.00     0.39   1.00 1827.2083
85001   600   10    4.00  21.58    10.900   47.088 1440.00     0.06   1.00 1827.2500
85001   700   10  -99.00  21.68    10.700   46.224 1462.00     0.06   1.00 1827.2917
85001   800   10    4.30  21.77    10.800   46.656 1480.00     0.05   1.00 1827.3333
85001   900   10    4.80  21.28    10.500   45.360 1506.00     0.07   1.00 1827.3750
```

```
Log of Archive, version: 1.4 of 1 June 1987,      processed on  9 Jun 1987

Files:

        RAW DATA demo9.raw
   CHANNEL TABLE demo.cht
         ARCHIVE demo9.ach
             LOG demo9.log

Archive delimiter is " ".
Missing or bad data values are replaced by the value -99.000 .

Line errors: are identified by their line number in the raw data file.
Data errors: are identified by the channels name, line and position
             within the case: "name  "(line in case/position in line).
Line numbers in raw data file are shown as |number| or as *number*
             |numbers| indicates a line of data, *numbers* is a comment line.

First case on raw data: 85 001 01:00
--------------------------------------------------------------------------------
BeginDate: 85 001 01:00                          First output case: 85 001 01:00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

*5* warning temperature T2 high

Data Error:
|6| 1  85001  300  10
|7| 2  1.7  216  10.1  1252.0
Value out of bounds: "T2      "(2/2);

Data Error:
|14| 1  85001  700  10
|15| 2  A4.2 22.4  10.7  1462.0
bad number: "T1      "(2/1);

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
EndDate:   99 365 24:00                           Last output case: 85 001 09:00
--------------------------------------------------------------------------------

STATISTICS:

  19 lines read from beginning of raw data file.
  19 lines processed between Begin and End dates.
    (including 1 comments and 0 all-blank lines)
   0 line errors detected.

   9 cases read; 9 cases archived.

   2 data errors, and 2 missing data detected, itemized below:

  Arc  Channel   Error  Missing
  pos   name     count   count
   4   T1          1       0
   5   T2          1       0
   7   H2O vol     0       1
   9   gas         0       1
```

I.3. Definitions

This section defines some of the terminology used in the remainder of the document.

Channel A single "variable" or "column". Typically, this is the output of one sensor, written to the raw data file once each sampling period.

Case (or record) A set of channels collected at one point in time. Each case must have a date and/or time value associated with it. A case can have several lines in the raw data file (multiple-line input case), but will always be written on one line in the archive file.

Line A line of the raw data file is a series of characters terminated by an end-of-line marker (CR and/or LF on ASCII systems). If an input case has multiple-line input, the lines may have a within-case line number used to distinguish them.

Missing value A value reserved to represent data which are absent, improperly formatted, or outside the checking limits. The value -99.00 is used by default. The missing value can be changed either by the program constant "MissingReal", or by a user-defined string (see Section III.2, flag 5).

I.4 Limits

Archive is not limited in the total length of the file (except in that it has an integer line counter which on some computers would limit the length to 32,767 lines). However, there are limits on the types of data it can process and what it can do with them.

Archive can process only a series of numbers organized into "cases." It cannot handle alphabetic data. (Text is treated as comment and written to the log file.) Input and output of numbers can be integer or real, and are stored internally as real numbers. Reals must be in floating point notation, not scientific notation (thus 4600 and 4600.00 are permitted, not 4.6E03). The largest integer that can be written to the archive file is 10,000 times the maximum integer number permitted by the computer in use (e.g., 327,670,000 is acceptable, even for a 16-bit micro-computer).

Archive can handle multiple-line input cases, but its archive file output is always one line per case (this could be changed by simple reprogramming, but we desired it for standardization). Archive cannot process more than one type of case within one file. An example of multiple types of cases would be hourly readings interspersed with ten-minute readings, each with a different number of channels. (Archive can handle this diversity only if each type is in a different data file.)

The channel table and raw data can have dates in either Gregorian or Julian format. Day, month, and year can be in any order in the raw data file, but must use the same delimiter as the rest of the data and must be on the first line of the case. Archive does not divide time more finely than one minute, although this would not be difficult for a programmer to add to the program.

The channel table is easily changed to add or rearrange channels, or to set a channel to "missing" for a given period of time. However, no provision is made for reducing the number of channels already defined. Unless a new channel table is written for the reduced set of channels, they will be reported as errors.

Program limits such as the maximum number of channels or characters per line can be altered by a very simple change to the Archive program itself.

Such limits are noted throughout the documentation; they involve what are called "program constants". Program constants appear in the first two pages of the program listing, and can be changed even by someone with little familiarity with the Pascal programming language.

II. RAW DATA

II.1 Cases

Each case in a raw data file consists of a set of data values including the date and/or time the case was recorded. Cases may be written on a single line or on several lines. The default maximum number of characters per raw line is 350; it can be increased by changing the program constant "MaxChrPerLine". Extra characters are ignored and an error message is written to the log file. The data values may be separated by space or comma[1], or by the '+' or '-' sign at the beginning of a new data value. Each data value in the case corresponds to a channel of the data acquisition system and is identified by its position in the case (e.g., line number and position in the line).

If multiple-line cases are used, lines can optionally have line numbers which will be used to check for "alignment" or completeness of the case[2]. It is therefore strongly recommended that line numbers be included in any raw data having multiple lines per case. Such line numbers are <u>not</u> counted as channels, and therefore need not be identified in the channel table. Line numbers sometimes correspond to the board number of the data acquisition system, each board having a given number of channels. For Archive, line numbers do not need to be adjacent or sequential. For example, a three-line case with the sequence of line numbers 1, 11, 9 is acceptable. There is an upper limit of line numbers of 25 (changed by program constants "MaxRawLine"and "MaxRawQueue").

---

[1]By default, any number of commas and blanks between values are skipped. Alternatively, the comma may be a place holder, by setting the constant "CommaIsPlaceHolder" to true. (For instance, if true, a line containing "1,,2" would be read as three values: a 1, a missing, and a 2).

[2]Data with multiple lines per case and no line numbers will produce garbage once a line is missing, since data will erroneously be assigned to the wrong line positions. The date and time will likely be nonsense. A missing line is immediately recognized by the skyrocketing number of errors showing on the screen. If this happens, the raw data file must be edited and re-processed.

II.2 Date and Time

Program Archive uses the date and time of each case to govern its operation. Archive expects the date and time to be in the first line of each case, but within this line they can be positioned anywhere. The date can be Julian or Gregorian (see IV.1.3). The time must be in a twenty-four hour format (see IV.1.2). Other time and date formats are not accepted by Archive[3]. Date and time can be missing in the raw data, but many archive features will, as a result, not be available (see Appendix A.3).

II.3 Comment Lines

If more than 5% non-numeric characters appear on a line of the raw data file, it is considered a "comment line" (the percentage allowed is constant "MaxAlphaPct"). Comment lines are not written to the archive file, but are written to the screen and to the log file. They may be comments or warnings written to the raw data file by the data acquisition system or they may be non-numeric characters introduced by data transmission errors.

II.4 Examples of Raw Data

Following are several examples of raw data files which can be processed by Archive. More examples are shown in Appendix A.

1. Raw data file with one line per case and Julian date:

```
85001  100  10  1.4  21.2  10.0  1200.0
85001  200  12  1.6  21.5  9.8   1244.0
85001  300  10  1.7  21.3  10.2  1258.0
85001  400  10  1.6  21.8  10.5  1280.0
85001 1000   8  4.3  22.3  10.8  1420.0
85001 1100  10  4.8  22.0  10.5  1480.0
```

2. Same data file with multiple lines per case, without line numbers:

```
85001  100  10
1.4  21.2  10.0  1200.0
85001  200  12
1.6  21.5  9.8   1244.0
85001  300  10
1.7  21.3  10.2  1258.0
```

---

[3]Other formats could be accommodated with a small program change for time formats such as 09:00, or 09:00:00, or with other separators. Seconds are currently not included in the time processing routines.

```
85001  400  10
1.6  21.8  10.5  1280.0
85001 1000   8
4.3  22.3  10.8  1420.0
85001 1100  10
4.8  22.0  10.5  1480.0
```

3. Same data file with multiple lines per case with line numbers:

```
1  85001  100  10
2  1.4  21.2  10.0  1200.0
1  85001  200  12
2  1.6  21.5  9.8   1244.0
1  85001  300  10
2  1.7  21.3  10.2  1258.0
1  85001  400  10
2  1.6  21.8  10.5  1280.0
1  85001 1000   8
2  4.3  22.3  10.8  1420.0
1  85001 1100  10
2  4.8  22.0  10.5  1480.0
```

4. Same data file comma-separated, one line per case, and with a Gregorian date:

```
85,01,01,100,10,1.4,21.2,10.0,1200.0
85,01,01,200,12,1.6,21.5,9.8,1244.0
85,01,01,300,10,1.7,21.3,10.2,1258.0
85,01,01,400,10,1.6,21.8,10.5,1280.0
85,01,01,1000,8,4.3,22.3,10.8,1420.0
85,01,01,1100,10,4.8,22.0,10.5,1480.0
```

5. Raw data file produced by Campbell scientific CR-21:
(see also Appendix A.4).

```
*
*****E
*D
******E
*390D
01+0208.  02+4.000  03+0365.  04+2200.  05+44.95  06+17.58  07+18.51
08-4.302
09-6999.  10+3051.  11+2567.  12+2783.  13-155.5  14+0.520  15+0.480
16+1.000
17+0.000  18+1.000  19+0.000  20+0.298  21+50.38  22+36.66  23+0.000
01+0208.  02+4.000  03+0365.  04+2230.  05+44.91  06+17.86  07+18.77
08-4.558
09-6999.  10+2982.  11+2563.  12+2695.  13-146.7  14+0.337  15+0.663
16+1.000
17+0.000  18+1.000  19+0.000  20+0.297  21+47.69  22+36.06  23+0.000
```

```
01+0208.  02+4.000  03+0365.  04+2300.  05+44.90  06+18.02  07+19.12
08-4.353
09-6999.  10+3083.  11+2568.  12+2809.  13-163.7  14+0.580  15+0.420
16+1.000
17+0.000  18+1.000  19+0.000  20+0.296  21+47.92  22+38.56  23+0.000
01+0208.  02+4.000  03+0365.  04+2330.  05+45.11  06+17.64  07+18.97
08-3.282
09-6999.  10+3075.  11+2570.  12+2835.  13-163.2  14+0.630  15+0.370
16+1.000
17+0.000  18+1.000  19+0.000  20+0.295  21+47.79  22+36.66  23+0.000
01+0208.  02+4.000  03+0366.  04+0000.  05+45.15  06+17.76  07+18.76
08-1.843
09-6999.  10+2943.  11+2564.  12+2723.  13-147.1  14+0.380  15+0.620
16+1.000
17+0.000  18+1.000  19+0.000  20+0.300  21+48.52  22+36.70  23+0.000
```

II.5 Errors in the raw data file

Due to data transmission errors or power failures, cases may sometimes be incomplete. Archive will then issue an error message to the log file. Errors may be detected when first reading that data on an individual line ("line errors") or when converting, checking, and writing out the entire case ("data errors"). Line errors are written to the log file with just the one line on which the error occurred. Data errors include the entire case. For either error, the number of the line, counted from the beginning of the raw data file, indicates the place at which the error occured. The line number can be used to inspect and correct the case and line in the raw data file using a text editor.

II.5.1 Line errors

The following errors are considered line errors:

- the number of data values in a line of raw data is not equal to the number declared in the channel table;

- the line exceeds the maximum number of characters permitted (default is 350 which can be adjusted with the program constant "MaxChsPerLine");

- a data value exceeds the maximum number of characters permitted (default is 25 which can be adjusted by the program constant "MaxField"+1);

- for cases with line numbers: a line with a line number that is not declared in the channel table or a line whose line number appears for a second time in the currently processed case. (In either instance, the line will be ignored.)

In each case the faulty line is echoed to the log file with an appropriate error message. After the same error has occured 20 times (which is the current default -- it can be changed by the program constant

"MaxErrorCount"), a warning message is issued to the log file and error reporting is suppressed for this error.

II.5.2 Data errors

Reported data errors are:

- a data value is too big to write it in the specified integer format (see also Section V.);

- a data value is out of bounds after conversion has taken place (only appears when a bound check has been activated, see Section IV.2);

- a data value contains an unrecognizable character or two decimal points;

- special error checks (bound checks) are built in for time and date.

Every time an error is detected, the entire case is printed to the log file with an appropriate error message, indicating the position of the faulty data in the case and its line in the raw data file. After a channel has accumulated 20 errors (which is the current default that can be changed by the program constant "MaxErrorCount"), a warning message is issued to the log file and error reporting is suppressed for this error.

II.5.3 Example of a raw data file with errors

We give as an example an especially noisy data file, as might come through a modem over a bad phone line from the field. This file will be processed by Archive without problem, although some data are not recoverable. Italicized notes on the right describe what archive will do with the file. All errors and comments are reported to the log file.

RAW DATA FILE	ACTIONS TAKEN BY ARCHIVE
y	
@@@	*Comment lines written to*
4875 characters transmitted	
	screen and log file
data from field site XYZ	
1 85001 100 10	
2 1.4 21.2 10.0 1200.0	
1 85001 200 12	
2 1.6 21.5 9.8 1244.0	
	blank line skipped
alarm set high on channel 3	*comment line written to log file, screen*
	blank line skipped
1 85001 300 10	
2 1.7	*incomplete case, missing values inserted*
1 85001 400 10	
@@@@%%&&!@^	*comment line*
2 1.6 21.8 10.5 1280.0	
2 4.3 22.3 10.8 1420.0	*no line 1, no date, not processed*
1 85001 500 10	
2 4.322.3 10.8 1420	*bad number, replaced with missing*
1 85001 600 10.00000001	
2 4.3 22.3 10800000.5 1429.0	
1 85001 1100 10	*no line 2, line of missing values inserted*
2 4.8 22.0 lkj^&^& 10.5 1480.0	*comment line (non-numeric characters)*
^@^@	*comment line*
end communication	*comment line*

III. CHANNEL TABLE

The channel table is a file created by the user containing descriptions of all channels that appear in each case of the raw data file. It must be a plain ASCII file, like a file which can be produced with a text editor. Some word processors add control codes which will cause errors[4]. For quick production of new channel tables, a blank form of the channel table is provided with the Archive distribution disk, called Blank.cht. The channel table lines are long (typically over 80 columns), and it is important that the editor or word processor not "wrap" them; this may require adjusting the margins on some word processors.

The channel table is read by Archive, which extracts from it all the necessary information for checking and converting the raw data. If an error is detected while reading the channel table, a message is sent to the

---

[4]On microcomputers, be sure to produce a "non-document" file (Wordstar) or "DOS text" file (Text-out function in Wordperfect). It is advisable not to use the TAB key. On non-ASCII machines other character sets, such as EBCDIC, can be used with only minor programming changes to Archive.

screen, and the program terminates the run. (For finding errors in the channel table, see Section III.2 - flag 4.) In the following, the content of the channel table and its function are explained in detail. An example channel table is shown first below.

Example channel table for raw data of Section II.5:

Date MM/DD/YY	Time HH:mm	Raw line	Raw coln	Arch pos	Name of channel	Archive Units	Arch Format	Conv'n Code	Calib. or Constants	Error Code	Limitation Constants	Channel Description
#24												
01/01/85	01:00	1	1	1	JulDate	YYDDD	I5	20		0		
01/01/85	01:00	1	2	2	Time	HHmm	I5	15		0		
01/01/85	01:00	1	3	3	#polled	-	I4	1		0		# sampling
01/01/85	01:00	2	1	4	T1	deg C	F6.2	1		0		temperature
01/01/85	01:00	2	2	5	T2	deg C	F6.2	2	0.99 -0.5	1	0 200	temperature
01/01/85	01:00	2	3	6	flow	liter/s	F8.3	1		0		circulation
01/01/85	01:00	2	3	7	H2O vol	liter	F8.3	6	0.0012	1	0 2000	water use
01/01/85	01:00	2	4	8	gas	ft3	F7.2	1		0		gas meter
01/01/85	01:00	2	4	9	gas	ft3/s	F7.2	8	10 20000	1	0 200	gas usage
01/01/85	01:00	1	1	10	—	hours	F5.2	18		0	(virtual: time-gap)	
01/01/85	01:00	1	1	11	—	days	F9.4	28		0	(virtual: "decimal time")	

III.1 Heading

The first few lines in the example channel table are optional and can be used for a description of the experiment and column identification. The heading may contain any amount of text. It is treated as a "comment" in the channel table file, and is not processed by Archive. It usually also includes column identification to improve readability. The heading is terminated by the "flag line" which begins with a pound sign "#"; thus the pound sign may not appear as the first character in any line of the heading.

III.2 Flag line

The flag line is required and must begin with a "#". It is followed optionally by option "flags", of which five are currently recognized:

2 Line numbers in raw data for multiple-line cases (default is no line numbers).
3 Julian dates in channel table, (Gregorian is default).
4 Echo channel table to screen as it is being read (useful when first testing a new channel table).
5 User-defined missing identifier (default is -99.0).
6 User-defined archive delimiter (default is blank).

The flags may be separated by blanks. They are described in more detail below:

Flag "2" means that there are multiple lines per case, and line numbers are present in the data cases (at the beginning of each line of the case). If the cases of raw data consist of one line only then the user should leave the flag at its default setting.

Flag "3" means that the dates in the channel table are written as Julian dates (e.g., 86 234). The blank between the year and day is necessary. The default is the Gregorian date (e.g., 10/27/86). Notice that this is specified independently of the date format in the raw data file.

Flag "4" means that the channel table is echoed to the screen as Archive reads it. This is a desirable debugging feature when errors appear due to improperly written channel tables. Once the channel table is correct, this flag should be turned off, since error messages may be lost in the volume of echoed channel table information.

Flag "5" can be used to specify a string (in parenthesis) which will be put in the Archive file for missing data. It need not be a number. The string will use as many spaces as defined by the field width (Section III.3.7.), using right justification, or, if longer, as many spaces as the defined string requires. Unlike the default missing value, a user defined missing string cannot be reformatted. For example, with 5(-999) a missing data value will be printed in F7.2 format as "   -999" not as "-999.00", and 5(*) will appear with format F7.2 as "   *". The default maximum number of characters is 12 and can be changed by program constant "MaxMissingStr".

Flag "6" and any one character in parenthesis allows free choice of a delimiter on the archive file. The default is a blank separator. For comma separation, for example, use 6(,). Separation by blanks is recommended, because some auxiliary analysis tools may require it.

Example flag line: #245(***)6(,)

III.3 Description of columns

The channel table is divided into 13 columns (some of which may have several values). The columns must be separated by at least one blank. Additional leading blanks before columns are ignored. The columns do not necessarily need to be aligned.

III.3.1 Date and time columns

The date and time must be given on every line of the table; they indicate the starting point of validity of the information on that line. Initially, they are the date and time at which the experiment was started; however, later they can be the date and time at which some change was made (adjusting of calibration constant, change of sensor location, etc.).

The date may be Gregorian, with month, day, and year, or Julian, with year and day, all separated by any single separator character. They must be in that order. The time must be expressed as a one or two-digit hour followed by a one-character separator, then a two-digit minute. Examples:

05/27/86 08:30
12-1-86 8-30
86 335 08:30 (Julian date, requires flag "3")

If year or date (but not both) is absent from the raw data file, conversion codes can be used to add a constant offset to the raw data, so that the channel table and archive file contains the true date and time (see Section IV.1.2 and IV.1.3). It is possible but less desirable to enter a year or date of zero on the channel table corresponding to the absent year or date on the raw file.

If neither time nor date is present in the raw data file, enter zeros for year, day, hour, and minute (see example in Appendix A.3). Conversion codes which require the time difference between cases will then be unavailable. Furthermore, updating of channels during a monitoring period will be impossible (unless a separate channel table file is used).

III.3.2 Raw line position

The line position within a case must always be indicated for each channel. If there is only one unnumbered line per case, each channel table entry must nevertheless enter a line position of "1". For multiple lines per case, the line numbers in the raw data case are used if present, otherwise line numbers are counted sequentially starting with 1. Line numbers need not be in sequence either in the channel table or the raw data file. The default range of allowed line numbers is 0 to 25. (This can be changed with the program constants "MaxRawLine" and "MaxRawQueue".)

III.3.3 Raw column position

The raw column position indicates the position of the channel within the raw line. If line numbers are present, the line number occupies raw position "0" and is not counted as a channel and does not need to be specified (although it may be if that is desirable.)

Channels that exist in the raw data but do not need to be archived may be skipped by not declaring their position in the channel table. However, for clear error reports, the last channel on every line should be declared. If it is an undesired channel, it can be discarded by assigning archive position "0" to it. (Failing to declare the last channel results in a warning message which reports that more channels are found on a line than were expected.)

The range of raw positions is 0 to 40 (which can be changed by the program constant "MaxChannelsPerLine"). It is not necessary to declare channels in sequence.

III.3.4 Archive position

After conversion and checking, each case is written to the archive file as a series of numbers on a single line. The archive position is the position to which the data for the given channel will be written. The range of archive positions is 1 to 120 (program constant "MaxChannels").

An archive position of "0" indicates that data for the channel should not be written to the archive file. This can be used to discard unwanted data in the raw data file.

Unfilled archive positions: An archive position may be left open for later use. This is handy if channels are to be added later, yet a consistant format is desired for the entire monitoring period. The undeclared position will be filled with "missing values" and a warning message will appear while the channel table is being read.

For intensive users of Archive, it may be useful to know that the writing of the archive file is "driven" by archive positions, not raw positions. For each line of output, the program steps through each archive position, performs any necessary checking and conversion, and writes the datum. This explains an assymetry in updating the channel table: A new entry with a changed raw channel for the same archive position will supersede the old entry. But a new entry with a changed archive position for the same raw channel will cause the same raw datum to be written to both the old and the new archive positions. (See conversion code 0, Section IV.1.1: this code will cancel writing of a raw data value to an archive position by putting in a missing value.)

III.3.5 Channel name

The channel name serves as additional descriptive identification of each channel. A fixed field of eight characters counted from the first non-blank character is reserved for the name (program constant "CharsPerChName"). This field cannot be left entirely blank, since leading blanks are skipped and the next field would be mistaken for the channel name. If no name is specified, the user should enter "-" followed by at least seven blanks ("CharsPerChName"-1 blanks).

"End" and "Begin" are reserved names (upper or lower case). They indicate the beginning and end of an experiment, and are used to skip raw data that is not to be processed (See Section III.4.4).

III.3.6 Archive units

The archive units serve as documentation for the user of the units to which the data were converted when written to the archive file. A fixed field of twelve characters counted from the first non-blank character is reserved for the units (program constant "CharsPerUnit"). This field cannot be left entirely blank. If no units are specified, enter "-" followed by at least 11 blanks ("CharsPerUnit-1 blanks).

III.3.7 Archive format

The format in which the data are written to the archive file is specified in this column. It can be either integer or real format. The convention used here is that of Fortran. "I" or "i" stands for integer format, "F" or "f" for real format. The next one or two-digit number number after the format specifier indicates the entire field width. No space is permitted between format specifier and field width. In the case of the real format a decimal point is required followed by the field width of the decimal part.

Examples: I4, i20, F7.2, f1.2, F50.10

The delimiter is not part of the field width. Should a value need more spaces than available from the field width, more spaces will be used. For example,

```
field specification:   F7.3        I3
raw data:
10                  | 10.000|    | 10|
10000               |10000.000   |10000
0.001               |  0.001|    |  0|
0.00001             |  0.000|    |  0|
```

Minimal space for the archive file is obtained by specifying a field width of 1, which uses the minimum number of characters necessary to write the number. However, column alignment will probably be lost.

III.3.8 Conversion code

The conversion code is an integer between 0 and 31 (the maximum number of existing conversion codes), and must be specified. Conversion codes are available for various algebraic transformations, including identity. All conversion codes are described in Section IV.

III.3.9 Conversion constants

The conversion code automatically determines how many calibration or conversion constants have to be specified (see Section IV). Archive will skip to the next column if no conversion constants are required, or will read the necessary number of constants. The constants may be integer or real format, and must be separated by blanks.

Warning: A mismatch between the number of constants entered and the number required by the conversion code will result in an error which may not necessarily be detected by the program and which could result in erroneous data in the archive file.

III.3.10 Error checking code

An error checking code must be specified. The current version of Archive provides only two options: 0 (no check) or 1 (high-low bound check), as described in Section IV.2.

III.3.11 Error checking constants

The error checking code automatically determines how many error checking constants have to be specified (see Section IV). Archive will move on to the next line if no error checking constants are called for, or will read the necessary number of constants. The constants may be in integer or real format. They are separated by blanks.

III.3.12 Channel description

Text can be added at the end of the line. It is ignored by Archive, but can be very helpful for documentation.

III.4 Updating the channel table

The channel table can be conveniently updated without interrupting the flow of data, simply by adding a line at the end which redefines the channel and specifies the time and date when the change took place in the field. Archive will process all raw data up to the date the change was marked, then read the new channel table entry (or entries) and then continue processing raw data until a new date on the channel table indicates a further update.

When updating involves channels that use the preceding raw datum, (for example, channels that use conversion code 7 which computes a difference from the previous meter reading) they are reset.

III.5 "Begin" and "End" lines, delimiting experiments

"Begin" and "End" lines determine the time range of an experiment. They are used to include (or exclude) data during a specified time range, for example, when the instruments are used to conduct a temporary experiment during a long-term monitoring project. Begin and End lines are not needed for simply updating or adding channels. There is an implicit Begin prior to the first channel table entry, and an implicit End after the last one. Thus, if no short-term experiments are conducted, no Begin or End lines are required.

At the Begin, all counters are zeroed, and there is no possible comparison between data across the boundaries defined by Begin or End. This applies especially for rates or time differences.

III.5.1 Begin line

The Begin line identifies the starting date and time (inclusive) of the experiment, or the starting date and time of the particular configuration described by subsequent lines with the same time. The start of an experiment is indicated by "Begin" for the channel name (upper or lower case is acceptable), with sufficient number of blanks following to fill the fixed field of channel name characters (see Section III.3.5).

The Begin date need not coincide with the first date in the raw data file. If the Begin date precedes (comes before) the first date on the raw data file, the archiving process starts with the first case. If the Begin date is succeedes (comes after) the first date on the raw data file, cases will be skipped until the date on the raw data file reaches the Begin date. This synchronization requires that the date and time are at the same positions in the entire raw data file.

III.5.2 End line

The End line identifies the last date and time (inclusive) for which the configuration described by the preceding lines applies to the raw data. The end of an experiment is indicated by "End" for the channel name (upper or lower case is acceptable), with sufficient number of blanks followed to fill the fixed field of channel name characters (see Section III.3.5).

Raw data with later dates will not be processed or copied to the archive file. The End line can be omitted or the date can be set to a date far in the future for a complete processing of the raw data file.

The raw line, raw column, and archive columns are ignored; zeros can be used to distinguish Begin and End lines. For example:

```
86 208  8:30   0  0  0  Begin  (this space may be used for comments)
99 1   24:00   0  0  0  End
```

III.5.3 Skipping data

It is sometimes useful to skip data in the raw data file for a specific period of time. This is accomplished with the "Begin" and "End" lines, for example:

```
86 208 8:30  0  0  0  Begin
...
(channel table entries valid between 8:30 and 12:00, inclusive)
...
86 208 12:00 0  0  0  End
86 208 21:00 0  0  0  Begin
86 212 24:00 0  0  0  End
```

The data between 12:01 and 20:59 on day 208 are skipped. Note also that if no new entries are made between the second "Begin" and the second "End", the channel configuration remains the same as for the first chunk of data. The date and time positions in the skipped portion must not be changed, but other raw data positions or formats may be.

IV. CONVERSION CODES

This section describes the data conversions available in the Archive program. Conversion code numbering is organized as follows:

Code numbers	Type of data converted
0 to 14	raw data
15 to 19	time of day
20 to 29	year and date
30 up	raw data (customized)

The following abbreviations are used to describe the codes:

c_i conversion constants

x input (raw) data

x_{prev} input (raw) data from the case immediately preceding x

t, t_{prev} time of x and x_{prev}, respectively

t_{ref} Archive internal reference time

YY,MM,DD,HH,mm

two digit year, month, Gregorian day, hour, minute

DDD three digit (Julian) day

missing special missing value

For example; conversion code 2 is described as: "$c_1x + c_2$", which is a linear conversion. The input data value, x, is multiplied by the first constant, a second constant is added, and the result is stored in the archive file. The two constants must be given in the channel table following the "2" for conversion code 2. If the input value was missing, no conversion will occur; a missing value will be put in the archive file.

IV.1 Conversion codes

IV.1.1 Raw data conversion codes

0: missing — put missing value in archive position regardless of raw data

1: x — identity (no conversion)

2: $c_1x + c_2$ — linear transformation

3: $c_1 \ln(x) + c_2$ natural log (a check for $x > 0$ is included)

4: $c_1 \exp(x) + c_2$ exponential

5: $c_1+c_2x+c_3x^2+c_4x^3+c_5x^4$

polynomial

6: $c_1 x(t-t_{prev})$ integrated (dimension of x in units/seconds)

7: $c_1(x-x_{prev})$ difference for meter readings with wrap-around:

If the result is <0, the output is: $c_1(x - x_{prev} + c_2)$

c_2 is the wrap-constant and must be specified.
(If there is no wrap, set $c_2 = 0$).

8: $c_1(x-x_{prev})/(t-t_{prev})$

rate in units per second for meter readings with wrap:

if the result is <0, the output is: $c_1(x - x_{prev} + c_2)/(t-t_{prev})$

c_2 is the wrap-constant and must be specified.
(if there is no wrap, set $c_2 = 0$).

9: $(c_1x+c_2)/(t-t_{prev})$

rate in units per second

10 - 14: reserved

IV.1.2 Time of day conversion code

If time of day is to be used in the archiving process, it has to be identified by code 15 or 16. If it is only copied (say, with conversion code 1), Archive will not know to use it as the time of day.

15: x HHmm --> HHmm

Time of day, combined hours and minutes.
Output is same as input. Raw data format is HHmm e.g., 800 or 0800, 1400. Archive output format is HHmm e.g., 800, 1400 (but not 0800). Error check is built in: test for $0 <= x <= 2400$.

16: x and $x(c_1)$ HH mm --> HHmm

Time of day, separate hours and minutes.
Same as 15, but time format on raw data is HH mm, while archive output format will be HHmm. The raw column gives the position of hours, c_1 the position of minutes.

If it is desired to have the time in the archive output also separately (HH mm), specify "0" for Archive position to discard (HHmm) and add two more entries in the channel table to copy the hour and minutes separately, specifying conversion "1".

18: $t\text{-}t_{prev}$ Time interval between successive data samples.

Interval in hours and fractions thereof. The interval can be used only if either 15 or 16 is specified. Since the input to this conversion is the internal time, the entries for raw line and raw column are ignored; use any existing channel.

IV.1.3 Year and date conversion codes

For all year and date processing, offsets (if any) are added and year-end processing is done before Archive uses the date. Thus dates on the channel table file and archive file will always be true, corrected dates. Year-end processing, for example, would make year 1986, day 367, become year 1987, day 2. Leap years are correctly processed through year 2000.

Dates are stored internally in Julian form. Note that several of the date conversions allow offsets - these can be especially useful for recording correct year and Julian date in an archive file even if the field instrument only records sequential days since start of operation.

Julian formats:

20: x YYDDD --> YYDDD

Combined year and Julian date.
Output is same as input, Raw data and archive output format is YYDDD (or YYYYDDD), e.g., 86032 (or 1986032). Error check is built in: test for 0 <= YYYY <= 2050 and test for 1 <= DDD <= 366.

21: x YYDDD --> $(YY+c_1)(DDD+c_2)$ --> YYDDD

Combined year and Julian date with offset; c_1, c_2.
The raw column position is Julian date. An offset can be used to compensate for missing year (c_1) or date (c_2). Example: "86 0" would add 86 to the year and zero to the Julian day.

22: x and $x(c_1)$ YY DDD --> YYDDD

Separate year and Julian date.
The date format on raw data is YY DDD (separated), while Archive output format is YYDDD. The raw column gives year position, c_1 gives day position.

23: x and x(c_1)   YY DDD --> (YY+c2)(DDD+c3) --> YYDDD

Separate year and Julian date with offset; c_2, c_3.
The raw column is year position, c_1 is day position, c_2 is year offset, c_3 is day offset.

Gregorian format:

24: x, x(c_1), and xc_2

YY MM DD --> YYDDD   (or MM DD YY --> YYDDD, etc.)

Gregorian YY MM DD.
The raw column is year position, c_1 is month position, c_2 is day position. Output to Archive is YYDDD. No offsets available here.

IV.1.4 Decimal time conversion

Many types of analysis, especially graphical presentations, require the variable "time" to be continuous. However, our calendrical and chronometric systems are non-decimal at all levels: years, months, days, hours, and minutes. Thus, Archive computes a linear, decimal time which can be output to the archive file for later use. Typically, it would be output in addition to the standard-format time. The decimal time is computed with respect to the reference date of 1 January 1980; another reference date is easily specified by changing three program constants "RefYear", "RefDate", and "RefTod".

28: $t-t_{ref}$ Decimal time (output only).

Decimal time since January 1, 1980. Output is in days and fractions thereof. Since this conversion is based on the internal date and time, it will give a proper value only if date and time have been specified. The entries for raw line and raw column are ignored; use any existing channel (e.g., the entries for the date or time).

IV.1.5 User customized conversions:

Conversion codes of 30 and above are reserved for custom conversions which may not be included on the distributed versions of the Archive program. Users may modify the program code to implement their own conversions. To add conversion codes, users will need to make changes in the procedures ReadCht (case ConvCode of), EchoCht, Convert (case ConvCode of), and in the program constant "MaxConvCode".

IV.2 <u>Error checking codes</u>

0: x — No error check performed.

1: x — Bound check on input x performed:
c_1 <- x <- c_2

V. ARCHIVE FILE

When processed by Archive, each case of raw data becomes a single line of data values on the Archive file, using the conversions and error checking procedures for each channel. The data separator for the archive file will appear in addition to the field width specified by the format. If a value requires a field width larger than specified, more space will be used to accommodate the larger number (see Section III.3.7). Integers may be written up to 10,000 times the maximum integer value, in integer format (see Section I.4). All numbers are stored internally as reals.

There exists neither a limitation on the length of the lines nor on the length of the file that can be created, other than hardware limitations.

Below is an archive file produced from the raw data file, first shown in Section I.2:

85001	100	10	1.40	20.49	10.000	-99.000	1200.00	-99.00	0.00	1827.0417
85001	200	12	1.60	20.78	9.800	42.336	1244.00	0.12	1.00	1827.0833
85001	300	10	1.70	-99.00	10.100	43.632	1252.00	0.02	1.00	1827.1250
85001	400	10	1.60	21.08	10.500	45.360	1280.00	0.08	1.00	1827.1667
85001	500	10	4.30	21.38	10.800	46.656	1420.00	0.39	1.00	1827.2083
85001	600	10	4.00	21.58	10.900	47.088	1440.00	0.06	1.00	1827.2500
85001	700	10	-99.00	21.68	10.700	46.224	1462.00	0.06	1.00	1827.2917
85001	800	10	4.30	21.77	10.800	46.656	1480.00	0.05	1.00	1827.3333
85001	900	10	4.80	21.28	10.500	45.360	1506.00	0.07	1.00	1827.3750

VI. LOG FILE AND ERROR RECORDING

The log file keeps a record of what has been done during the archiving process. For convenience, the log file first shown in Section I.2 is reproduced on the following page.

A header includes the version of Archive in use and the date that Archive was run. All four file names are listed. The chosen Archive delimiter, the character separating data values in the archive file, and the string or value for bad or missing data are indicated. A short description explains the notation for line and data errors in the log file. If there is text on the raw data file prior to the first case of data, it will be shown as comment lines after the headings. This text may be an identification of the raw data file or other mesages.

The first readable time and date on the raw data file is called "First record on raw data". Following this, dashed lines separate "experiments". The variable "BeginDate" corresponds to the "Begin" date of the channel table and the variable "EndDate" to the "End" date of the channel table. Next to these two dates are written the dates of the first and the last output case that is written to the archive file. In between these dates, marked by the dashed lines, are recorded all comment lines and errors as they occur during the processing. Each line is preceded by a line number, with "*" surrounding line numbers of comment lines and "|" indicating error lines. The line number is counted from the very beginning of the raw data file; it is intended to facilitate locating the error on the raw data file with a text editor.

Line errors are recorded by printing the entire line to the log file. Line errors apply to the first decoding step, and involve: wrong number of channels, an invalid or duplicated line number within a case, or an excessively long number sequence.

Data errors are caused by single channels, due to missing values or a range check. For data errors, the entire case is reproduced as it appears on the raw data file. The case is written only once, even when several data errors are encountered in the same case. Short messages describe each error and where in the case it occurred.

If a data error on a particular channel appears repeatedly, the error reporting will be interrupted after reporting the same error 20 times (changed by the program constant "MaxErrorCount"). This prevents a single bad sensor from causing the entire raw data file to be written on the log file. Note that the error count is kept for each channel - thus if another sensor goes bad, writing of bad lines to the log file will resume. The same limit is applied to repeated line errors of the same type.

Finally an error statistic is included. A distinction is made between missing values and errors. A missing value can occur at the beginning of the file when time differences between the current and previous case are needed such as for determination of rates or differences. The first number will then be set to missing. A channel can also be deliberately set to missing (using conversion code 0).

```
Log of Archive, version: 1.4 of 1 June 1987,      processed on  9 Jun 1987

Files:

        RAW DATA demo9.raw
   CHANNEL TABLE demo.cht
         ARCHIVE demo9.ach
             LOG demo9.log

Archive delimiter is " ".
Missing or bad data values are replaced by the value -99.000 .

Line errors: are identified by their line number in the raw data file.
Data errors: are identified by the channels name, line and position
             within the case: "name  "(line in case/position in line).
Line numbers in raw data file are shown as |number| or as *number*
             |numbers| indicates a line of data, *numbers* is a comment line.

First case on raw data: 85 001 01:00
-------------------------------------------------------------------------------
BeginDate: 85 001 01:00                          First output case: 85 001 01:00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*5* warning temperature T2 high

Data Error:
|6| 1  85001  300  10
|7| 2  1.7  216  10.1  1252.0
Value out of bounds: "T2      "(2/2);

Data Error:
|14| 1  85001  700  10
|15| 2  A4.2 22.4  10.7  1462.0
bad number: "T1      "(2/1);
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
EndDate:   99 365 24:00                           Last output case: 85 001 09:00
-------------------------------------------------------------------------------

STATISTICS:

  19 lines read from beginning of raw data file.
  19 lines processed between Begin and End dates.
    (including 1 comments and 0 all-blank lines)
   0 line errors detected.

   9 cases read; 9 cases archived.

   2 data errors, and 2 missing data detected, itemized below:

  Arc  Channel   Error  Missing
  pos   name     count   count
   4   T1          1       0
   5   T2          1       0
   7   H20 vol     0       1
   9   gas         0       1
```

VII. RUNNING ARCHIVE

When running the program, the raw data file and the channel table file (in that order) have to be specified. This section describes how Archive is called from DOS or CP/M-86 systems. Details may differ slightly on other systems. The files can be given on the command line; e.g.,

```
ARCHIVE A:RAW.DAT B:CHANNEL.TBL
```

the program will then automatically assign default names using the file name of the raw data file. In this example, it would use the names A:RAW.ACH for the Archive file, and A:RAW.LOG for the log file (note that the same device or directory is used). If it is desired that the output files have different names or that they should be output to another output device or directory, then the two files can be specified fully on the command line, such as

```
ARCHIVE A:RAW.DAT B:CHANNEL.TBL B:FILE.ACH C:FILE.LOG
```

If no files are specified on the command line, then the program will prompt for the input files.

While running the program you will see a title on the screen indicating which version of Archive is running. Then a message appears that archive is reading the channel table. If flag 4 was given, the channel table is echoed onto the screen. The "end of channel table entry" indicates that the program has read the channel table until the next change in its date and will now proceed to process the data. If there are no raw data older than this last channel table date, the program will resume reading the channel table.

If there are raw data older than the "Begin" date on the channel table, the raw data will be skipped until the begin date is reached or surpassed, and data processing will start. While raw data are being skipped, the date and time of each case is shown on the screen together with the word "skipping".

While processing data, the date and time of each case are displayed together with an error counter that indicates the accumulated number of errors. When the "End" date is reached, a summary error statistic is written to the Log file and to the screen.

If there are more data on the raw data file, and if there are more entries on the channel table, channel table reading and raw data processing is resumed, ending when either the end of the raw data file or of the channel table file is reached. This is indicated on the screen with the message "ARCHIVE COMPLETE".

VIII. HINTS ON USING THE BOUNDS CHECK

Any use of archive must begin with a systematic description of the data to be collected. This description will become the channel table. The step of creating this description must be performed, or at least approved, by research staff familiar with the substance of the research, because they must establish the limits within which each channel will be considered "valid" data. (Archive allows you to skip the check for valid data, but we do not recommend this.)

When first setting up limits, it may be advisible to err on the side of accepting too much. For example, to measure outside temperature in Florida, you might accept any values between -50 F and 150 F. The reason for this is that you never know when that record-breaking cold wave will happen to hit during your study, and most actual failures will be wildly outside the measurement range (say -1000 F). After some familiarity with the data, one can go back and narrow the limits of acceptability. Conversely, as long as the raw data files have been preserved, one can also go back and specify broader limits, then re-run Archive.

IX. PROGRAMMING NOTES

The Archive source code is distributed on a microcomputer floppy diskette. It consists of about 2400 lines of Pascal. The main file, ARCHIVE.PAS, contains the declarations and statements of the main program, and the few procedures which need be changed when switching operating systems or computers. Three include directives access three files containing all additional procedures used by Archive.

Archive is written in Level 0 Standard Pascal as defined by ISO 7185. Further, it does not use procedural parameters, get/put, or file buffers. As a result, it can be compiled and run on most Pascal systems. At Princeton, we have tested it on Turbo Pascal, Prospero ProPascal, and BSD UNIX Pascal on computers ranging from an IBM-PC to a VAX. We are currently distributing it ready to be compiled by Turbo Pascal on MS-DOS. Conversion to other systems is quite simple. To illustrate the conversion process, the following steps were followed to compile Archive under UNIX on a minicomputer:

1. transfer files from PC to the UNIX system using error-free protocol such as Kermit
2. rename the source codes files from .PAS to .p
(all following changes made to file archive.p)
3. un-comment the #include directives, making the pound sign the first character on those lines
4. comment-out the system-date fetch in procedure GetAndWriteDate
5. comment-out the command line get in GetSysCommLine (the ambitious could rebuild the command line using argc and argv, otherwise it prompts the user for files)
6. change "assign(f,fn)" to "reset(f,fn)" in PrepFile
7. comment-out the two close statements at the end of the main program (close is needed only for Turbo)
8. compile using -s or {$L } flags to tell the pc compiler not to distinguish upper- from lower-case characters in word-symbols

Once the files were transferred, the entire conversion process required about 10 minutes.

X. ACKNOWLEDGMENTS

Many people have contributed to the development of this software. John DeCicco wrote the original software specification. Willett Kempton did the algorithm and data structure design and Daniel Feuermann and Kempton did the coding. Researchers of the Buildings Group at the Center for Energy and Environmental Studies made many helpful suggestions and provided extensive testing; Joseph Spadaro in particular found several tricky bugs. We are grateful to Peg Pierce for a heroic job of typing the manual on her last few days at work.

This work was supported by the New Jersey Energy Conservation Laboratory, for which the funding agencies are the seven New Jersey gas and electric utilities and the New Jersey Department of Commerce. The software is based in part upon on earlier work by Kempton supported by the National Science Foundation (NSF BNS-82-10088).

APPENDIX A: EXAMPLES

This section gives several example uses of archive on diverse types of data files.

A.1 Use of "Begin" and "End" lines to skip portion of the raw data.

A.1.1 Skipping beginning portion of raw data.

Raw data file:

```
1 85001 100
2 121   122
1 85001 200
2 221   222
1 85001 300
2 321   322
```

Channel table:

```
#24
01/01/85 02:00  0  0  0  Begin    (Begin time is after first raw date) <--
01/01/85 02:00  1  1  1  JulDate  yyddd         I5    20   0
01/01/85 02:00  1  2  2  Time     hhmm          I5    15   0
01/01/85 02:00  2  1  3  T21      ttt           F5.1  1    0
01/01/85 02:00  2  2  4  T22      ttt           I5    1    0
02/01/85 02:00  0  0  0  End
```

Archive file:

```
85001   200 221.0   222
85001   300 321.0   322
```

A.1.2 Skipping a mid-range of raw data.

Raw data file:

```
1 85001 100
2 121   122
1 85001 200
2 221   222
1 85001 300
2 321   322
```

Channel table:

```
#24
01/01/85 01:00  0  0  0  Begin   SKIP SECOND CASE ON RAW DATA
01/01/85 01:00  1  1  1  JulDate  yyddd         I5    20   0
01/01/85 01:00  1  2  2  Time     hhmm          I5    15   0
01/01/85 01:00  2  1  3  T21      ttt           F5.1  1    0
01/01/85 01:00  2  2  4  T22      ttt           I5    1    0
01/01/85 01:30  0  0  0  End                                      <---
01/01/85 02:30  0  0  0  Begin    new experiment:                 <---
01/01/85 02:30  2  1  3  T21      ttt           I6    1    0  (new format)
01/02/85 00:00  0  0  0  End
```

Archive file:

```
85001   100 121.0   122
85001   300     321   322
```

Log file (Note: The Log file is divided into two separate "reports" defined by the Begin/End bounds)

```
Log of Archive, version: 1.4 of 1 June 1987,      processed on  9 Jun 1987
Files:
        RAW DATA a:raw0
   CHANNEL TABLE a:cht02
         ARCHIVE a:raw0.ach
             LOG a:raw0.log

Archive delimiter is " ".
Missing or bad data values are replaced by the value -99.000 .

Line errors: are identified by their line number in the raw data file.
Data errors: are identified by the channels name, line and position
             within the case: "name    "(line in case/position in line).
Line numbers in raw data file are shown as |number| or as *number*
             |numbers| indicates a line of data, *numbers* is a comment line.

First case on raw data: 85 001 01:00
--------------------------------------------------------------------------
BeginDate: 85 001 01:00                      First output case: 85 001 01:00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
EndDate:   85 001 01:30                       Last output case: 85 001 01:00
--------------------------------------------------------------------------

STATISTICS:

   3 lines read from beginning of raw data file.
   2 lines processed between Begin and End dates.
    (including 0 comments and 0 all-blank lines)
   0 line errors detected.

   1 cases read; 1 cases archived.
   0 data errors, and 0 missing data detected

--------------------------------------------------------------------------
BeginDate: 85 001 02:30                      First output case: 85 001 03:00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
EndDate:   85 002 00:00                       Last output case: 85 001 03:00
--------------------------------------------------------------------------

STATISTICS:

   7 lines read from beginning of raw data file.
   3 lines processed between Begin and End dates.
    (including 0 comments and 1 all-blank lines)
   0 line errors detected.

   1 cases read; 1 cases archived.
   0 data errors, and 0 missing data detected
```

A.2 Using several conversion codes for a variable on the raw data file.

Raw data file:

```
1 85001 100
2  21    1
1 85001 200
2  22    2
1 85001 300
2  23    3
1 85001 400
2  23    3
1 85001 500
2  22    2
```

Channel table:

```
#24
01/01/85 01:00  1  1  1   JulDate   yyddd        I5    20         0
01/01/85 01:00  1  2  2   Time      hhmm         i5    15         0
01/01/85 01:00  1  1  15  deltat    hours        f5.2  18         0
01/01/85 01:00  1  1  16  decimalt  day          f10.4 28         0
01/01/85 01:00  2  1  3   T21       ttt          f5.1  1          0
01/01/85 01:00  2  2  4   T22       ttt          I5    1          0
01/01/85 01:00  2  2  5   t         t            f8.2  0          0
01/01/85 01:00  2  2  6   t         t            f8.2  2  1 1     0
01/01/85 01:00  2  2  7   t         t            f8.2  3  2 2     0
01/01/85 01:00  2  0  8   line#     t            f8.2  4  3 3     0
01/01/85 01:00  2  2  9   t         t            f8.2  5  1 2 3 4 5    0
01/01/85 01:00  2  2  10  t         t            f8.2  6  1       0
01/01/85 01:00  2  2  11  t         t            f8.2  7  2 1000  0
01/01/85 01:00  2  2  12  t         t            f8.2  8  2 1000  0
01/01/85 01:00  2  2  13  t         t            f8.2  30         0
01/01/85 01:00  2  2  14  t         t            f8.2  31   1 90  0
01/01/85 01:00  0  0  0   End
01/01/85 04:00  0  0  0   Begin
01/01/85 06:00  0  0  0   End
```

Archive file:

```
85001 100 21.0 1 -99.00 2.00 2.00 25.17 15.00 -99.00 -99.00 -99.00 -17.22 -99.00 0.00 1827.0417
85001 400 23.0 3 -99.00 4.00 4.20 25.17 547.00 -99.00 -99.00 -99.00 -16.11 -99.00 0.00 1827.1667
85001 500 22.0 2 -99.00 3.00 3.39 25.17 129.00 7200.00 1998.00 0.55 -16.67 -1.00 1.00 1827.2083
```

A.3 Variations in date and time on raw data file.

A.3.1 No date nor time available on raw data, missing data indicated by "*"

Raw data file:

```
123  12
234  23
345  34
456  45
567  56
678  67
789  78
890  89
```

Channel table:

Date mm/dd/yy	Time hh:mm	Lin pos	Raw pos	Arch pos	Name of channel	Archive Units	Arch Format	Conv'n Code	Calib. or Constants	Error Code	Limitation Constants	Channel Description
#45(*)6(,)												
00 00 00	00 00	1	1	1	3digit#	arbitray	f6.2	1		1	0 500	
00 00 00	00 00	1	2	2	2digit#	no units	i4	1		0		

Archive file:

```
123.00,  12
234.00,  23
345.00,  34
456.00,  45
     *,  56
     *,  67
     *,  78
     *,  89
```

Log file:

```
Log of Archive, version: 1.4 of 1 June 1987,     processed on  9 Jun 1987

Files:

        RAW DATA a:notimedt.raw
   CHANNEL TABLE a:notimedt.cht
         ARCHIVE b:notimedt.ach
             LOG b:notimedt.log

Archive delimiter is ",".
Missing or bad data values are replaced by the string "*".

Line errors: are identified by their line number in the raw data file.
Data errors: are identified by the channels name, line and position
             within the case: "name   "(line in case/position in line).
Line numbers in raw data file are shown as |number| or as *number*
             |numbers| indicates a line of data, *numbers* is a comment line.


First case on raw data:  0 xxx 00:00

------------------------------------------------------------------------
BeginDate:  0 xxx 00:00                       First output case:  0 xxx 00:00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Data Error:
|5| 567  56
Value out of bounds: "3digit# "(1/1);

Data Error:
|6| 678  67
Value out of bounds: "3digit# "(1/1);
```

```
Data Error:
|7| 789  78
Value out of bounds: "3digit# "(1/1);

Data Error:
|8| 890  89
Value out of bounds: "3digit# "(1/1);

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
EndDate:   99 365 24:00                        Last output case:  0 xxx 00:00
--------------------------------------------------------------------------------

STATISTICS:

   9 lines read from beginning of raw data file.
   9 lines processed between Begin and End dates.
    (including 0 comments and 1 all-blank lines)
   0 line errors detected.


   4 data errors, and 0 missing data detected, itemized below:

  Arc  Channel   Error  Missing
  pos   name     count   count
   1   3digit#     4       0
```

A.3.2 <u>No year in raw data - conversion 21 (offset year=1987)</u>

Raw data file:

```
0 035 1000
1 11 12 13
2 21 22 23
0 035 1010
1 11 12 13
2 21 22 23
0 035 1020
1 11 12 13
2 21 22 23
```

Channel table:

```
#24
01/01/85 02:00  0  1  1  JulDate  yyddd        I5    21 1987 0   0 <---
01/01/85 02:00  0  2  2  Time     hhmm         I5    15          0
01/01/85 02:00  0  1  3  DecTime  DDD.fract    F10.4 28          0
01/01/85 02:00  0  1  4  Gap      hour.fract   F10.4 18          0
01/01/85 02:00  1  1  5  T11      ttt          i5    1           0
01/01/85 02:00  1  2  6  T12      ttt          i5    1           0
01/01/85 02:00  1  3  7  T13      ttt          i5    1           0
01/01/85 02:00  2  1  8  T21      ttt          F5.1  1           0
01/01/85 02:00  2  2  9  T22      ttt          I5    1           0
01/01/85 02:00  2  3 10  T23      ttt          i5    1           0
```

Archive file:

```
1987035  1000  2591.4167    0.0000   11    12    13  21.0    22    23
1987035  1010  2591.4236    0.1667   11    12    13  21.0    22    23
1987035  1020  2591.4306    0.1667   11    12    13  21.0    22    23
```

A.3.3 Year and Julian day separately on raw data - conversion 22
(day-position = 2).

Raw data file:

```
0 1987 035 1000
1 11 12 13
2 21 22 23
0 1987 035 1010
1 11 12 13
2 21 22 23
0 1987 035 1020
1 11 12 13
2 21 22 23
```

Channel table:

```
#24
01/01/85 02:00  0  1  1  JulDate  yyddd       I5    22  2    0
01/01/85 02:00  0  3  2  Time     HHmm        I5    15       0
01/01/85 02:00  0  1  3  DecTime  DDD.fract   F10.4 28       0
    .
    .
    .
```

Archive file:

```
1987035  1000  2591.4167    0.0000   11    12    13  21.0    22    23
1987035  1010  2591.4236    0.1667   11    12    13  21.0    22    23
1987035  1020  2591.4306    0.1667   11    12    13  21.0    22    23
```

A.3.4 Using offsets for day and year.

Raw data file:

```
0 1987 035 1000
1 11 12 13
2 21 22 23
0 1987 035 1010
1 11 12 13
2 21 22 23
0 1987 035 1020
1 11 12 13
2 21 22 23
```

Channel table:

```
#24
01/01/85 02:00  0  1  1  JulDate  yyddd        I5     23 2 -1 20  0  <---
01/01/85 02:00  0  3  2  Time     hhmm         I5     15          0
01/01/85 02:00  0  1  3  DecTime  DDD.fract    F10.4  28          0
01/01/85 02:00  0  1  4  Gap      hour.fract   F10.4  18          0
    .
    .
    .
```

Archive file:

```
1986055  1000  2246.4167   0.0000  11  12  13  21.0  22  23
1986055  1010  2246.4236   0.1667  11  12  13  21.0  22  23
1986055  1020  2246.4306   0.1667  11  12  13  21.0  22  23
```

A.3.5 Gregorian date on raw data, separate hour and minutes.

use conversion 24 (month position, day position)

use conversion 16 (minute position)

Raw data file:

```
0 1987 02 4 10 00
1 11 12 13
2 21 22 23
0 1987 02 4 10 10
1 11 12 13
2 21 22 23
0 1987 02 4 10 20
1 11 12 13
2 21 22 23
```

Channel table:

```
#24
01/01/85 02:00  0  1  1  JulDate  yyddd        I5     24  2  3    0  <---
01/01/85 02:00  0  4  2  Time     hhmm         I5     16  5       0  <---
01/01/85 02:00  0  1  3  DecTime  DDD.fract    F10.4  28          0
01/01/85 02:00  0  1  4  Gap      hour.fract   F10.4  18          0
    .
    .
    .
```

Archive file:

```
1987035  1000  2591.4167   0.0000  11  12  13  21.0  22  23
1987035  1010  2591.4236   0.1667  11  12  13  21.0  22  23
1987035  1020  2591.4306   0.1667  11  12  13  21.0  22  23
```

A.4 Campbell scientific CR-21 data format

Raw data file:

```
*
*****E
*D
******E
*390D
01+0208.  02+4.000  03+0365.  04+2200.  05+44.95  06+17.58  07+18.51  08-4.302
09-6999.  10+3051.  11+2567.  12+2783.  13-155.5  14+0.520  15+0.480  16+1.000
17+0.000  18+1.000  19+0.000  20+0.298  21+50.38  22+36.66  23+0.000
01+0208.  02+4.000  03+0365.  04+2230.  05+44.91  06+17.86  07+18.77  08-4.558
09-6999.  10+2982.  11+2563.  12+2695.  13-146.7  14+0.337  15+0.663  16+1.000
17+0.000  18+1.000  19+0.000  20+0.297  21+47.69  22+36.06  23+0.000
01+0208.  02+4.000  03+0365.  04+2300.  05+44.90  06+18.02  07+19.12  08-4.353
09-6999.  10+3083.  11+2568.  12+2809.  13-163.7  14+0.580  15+0.420  16+1.000
17+0.000  18+1.000  19+0.000  20+0.296  21+47.92  22+38.56  23+0.000
01+0208.  02+4.000  03+0365.  04+2330.  05+45.11  06+17.64  07+18.97  08-3.282
09-6999.  10+3075.  11+2570.  12+2835.  13-163.2  14+0.630  15+0.370  16+1.000
17+0.000  18+1.000  19+0.000  20+0.295  21+47.79  22+36.66  23+0.000
01+0208.  02+4.000  03+0366.  04+0000.  05+45.15  06+17.76  07+18.76  08-1.843
09-6999.  10+2943.  11+2564.  12+2723.  13-147.1  14+0.380  15+0.620  16+1.000
17+0.000  18+1.000  19+0.000  20+0.300  21+48.52  22+36.70  23+0.000
```

Channel table:

Note: Campbell CR-21 raw data files include a channel number in front of each data value. This channel table treats the first channel number on every line in the raw data file as a line number (1, 9, 17). The line number is considered the "zero" position. So the useful data have odd raw positions (1,3,5...). The channel numbers are simply being ignored. The same could be done without line numbers, but the data would be messed up if a line were missing.

The date is adjusted by a constant offset of 86 in the channel table with conversion code 21. The number "0" next to the offset is the offset for the day. Note that the archive file begins with 86DDD, then wraps to the next year 87DDD when DDD > 365 (it would have wrapped at 366, if 1986 had been a leap year).

Luciano's initial 10-days of data.

Date mm/dd/yy	Time hh:mm	Lin pos	Raw pos	Arc pos	Name of channel	Archive Units	Arch Format	Conv Code	Calib. or Constants	Err. Code	Limit. Const	Channel Description
#245(*)												
10/11/86	09:30	1	3	1	House#	unit	I1	1			0	
10/11/86	09:30	1	5	2	DaTE	DDD	I1	21	86 0	0		
10/11/86	09:30	1	7	3	Time	hh:mm	I1	15		0		
10/11/86	09:30	1	9	4	RH	%	f1.2	1		1	0 100	
10/11/86	09:30	1	11	5	Tbase	C	f1.2	1		1	-25 50	
10/11/86	09:30	1	13	6	Tfirst	C	f1.2	1		1	-25 50	
10/11/86	09:30	1	15	7	Tout	C	f1.2	1		1	-25 50	
10/11/86	09:30	9	1	8	Txtra	C	f1.2	1		0	-25 50	checking disabled
10/11/86	09:30	9	3	9	DP,b/o	mV	f1.2	1		1	0 5000	

```
10/11/86 09:30    9    5  10 DP,b/s    mV          f1.2     1                1    0  5000
10/11/86 09:30    9    7  11 DP,b/u    mV          f1.2     1                1    0  5000
10/11/86 09:30    9    9  12 DP,xtra   mV          f1.2     1                0    0  5000 checking disabled
10/11/86 09:30    9   11  13 SW1       +/-1        f1.2     1                1    0  1
10/11/86 09:30    9   15  14 SW2       +/-1        I1       1                0
10/11/86 09:30   17    3  15 SW3       +/-1        I1       1                0
10/11/86 09:30   17    7  16 SoMo      sh          f1.2     1                0
10/11/86 09:30   17    9  17 Rn,b      pC/l        f1.2     9  2147 -945     1  0 1000
10/11/86 09:30   17   11  18 Rn,up     pC/l        f1.2     9  2020 -1273    1  0 1000
10/11/86 09:30   17   13  19 Rn,xtr    pC/l        f1.2     1                1  0 1000
```

Archive file:

```
4 86365 2200 44.95 17.58 18.51 -4.30 -6999.00 3051.00 2567.00 2783.00 -155.50 0.52 1 1 0.30 * * 0.00
4 86365 2230 44.91 17.86 18.77 -4.56 -6999.00 2982.00 2563.00 2695.00 -146.70 0.34 1 1 0.30 56.36 39.76 0.00
4 86365 2300 44.90 18.02 19.12 -4.35 -6999.00 3083.00 2568.00 2809.00 -163.70 0.58 1 1 0.30 56.63 42.57 0.00
4 86365 2330 45.11 17.64 18.97 -3.28 -6999.00 3075.00 2570.00 2835.00 -163.20 0.63 1 1 0.29 56.48 40.43 0.00
4 87001 0 45.15 17.76 18.76 -1.84 -6999.00 2943.00 2564.00 2723.00 -147.10 0.38 1 1 0.30 57.35 40.48 0.00
```

Log file:

```
Log of Archive, version: 1.4 of 1 June 1987,     processed on  9 Jun 1987
Files:

        RAW DATA a:rn3.raw
   CHANNEL TABLE a:rn4.cht
         ARCHIVE a:rn3.ach
             LOG b:rn3.log

Archive delimiter is " ".
Missing or bad data values are replaced by the string "*".

Line errors: are identified by their line number in the raw data file.
Data errors: are identified by the channels name, line and position
             within the case: "name  "(line in case/position in line).
Line numbers in raw data file are shown as |number| or as *number*
             |numbers| indicates a line of data, *numbers* is a comment line.

*1* *

*2* *****E

*3* *D

*4* ******E

*5* *390D

First case on raw data: 86 365 22:00

--------------------------------------------------------------------------
BeginDate: 86 284 09:30                        First output case: 86 365 22:00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
EndDate:   99 365 24:00                         Last output case: 87 001 00:00
```

STATISTICS:

20 lines read from beginning of raw data file.
15 lines processed between Begin and End dates.
(including 0 comments and 0 all-blank lines)
0 line errors detected.

5 cases read; 5 cases archived.

0 data errors, and 2 missing data detected, itemized below:

Arc pos	Channel name	Error count	Missing count
17	Rn,b	0	1
18	Rn,up	0	1

A.5 Practical example for raw data without line numbers.

Raw data file:

```
86009 1120   0.85  85005 613553
86009 1140   1.39  85066 614045
86009 1200   1.94  85113 614250
86009 1240   2.48   0.00   0.00   1497  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
86009 1300   2.48   0.00 248.00  11656  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
86009 1320   3.02 158.00   1016  21807  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
86009 1340   3.57 383.00   1016  32029  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
86009 1400   3.57 686.00   1016  41963  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
86009 1440   0.00 686.00   1016  41963  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
```

Channel table:

Date mm/dd/yy	Time hh:mm	LINE pos	RAW pos	ACH pos	Name of channel	Archive Units	ACH Format	Conversion and calibration Code	Constants:		Error checking Code	Constants:		Channel Description
#														
01/09/86	12:20	1	0	0	Begin	Lumley TT12 monitoring								
01/09/86	12:20	1	1	1	Date	yyddd	I5	20			0			Julian date
01/09/86	12:20	1	2	2	Time	hh:mm	I5	15			0			time of day
01/09/86	12:20	1	0	3	Gap	hours	F8.2	18			0			time gap between polls
01/09/86	12:20	1	3	4	T-out	C	F6.1	1			1	-24	44	outdoor temperature
01/09/86	12:20	1	4	5	G-main	kW	F8.1	8	10810	1000000	1	0	800	main meter gas usage
01/09/86	12:20	1	5	6	G-cook	kW	F8.1	8	1081	1000000	1	0	100	cooking gas usage
01/09/86	12:20	1	6	7	Elect	kW	F8.1	8	180	1000000	1	0	100	electrical consumption
01/09/86	12:20	1	7	0	P-steam	psig	F8.1	0			1	0	15	steam pressure
01/09/86	12:20	1	8	0	Q-cond	l/s	F8.1	8	0.3875	100000	1	0	500	condensate flow
01/09/86	12:20	1	9	0	T-cond	C	F6.1	0			1	20	100	condensate temperature
01/09/86	12:20	1	10	8	T-hot	C	F6.1	2	0.5556	-3.116	1	10	90	hot water supply temp
01/09/86	12:20	1	11	0	T-cold	C	F6.1	0			1	0	40	cold water supply temp
01/09/86	12:20	1	12	9	T-ret	C	F6.1	2	0.5556	-3.390	1	10	90	DHW recirculation temp
01/09/86	12:20	1	13	0	Q-ret	l/s	F8.1	0			0			DHW recirculation flow
01/09/86	12:20	1	14	10	Q-cold	l/s	F8.1	8	38.75	1000000	1	0	100	DHW system makeup flow
01/09/86	12:20	1	1	11	RefTime	Day-Fraction	F10.4	28			0			
01/09/86	15:15	1	0	0	End									

Archive file:

```
86009  1240   0.00  2.5  -99.0  -99.0  -99.0 -99.0 -99.0  -99.0 2200.5278
86009  1300   0.33  2.5    0.0  -99.0  -99.0 -99.0 -99.0    0.0 2200.5417
86009  1320   0.33  3.0  -99.0  -99.0  -99.0 -99.0 -99.0    0.0 2200.5556
86009  1340   0.33  3.6  -99.0    0.0  -99.0 -99.0 -99.0    0.0 2200.5694
86009  1400   0.33  3.6  -99.0    0.0  -99.0 -99.0 -99.0    0.0 2200.5833
86009  1440   0.67  0.0    0.0    0.0    0.0 -99.0 -99.0    0.0 2200.6111
86009  1500   0.33  4.6  459.4  -99.0  -99.0 -99.0 -99.0    0.0 2200.6250
```

Log file:

```
Log of Archive, version: 1.4 of 1 June 1987,      processed on  9 Jun 1987

Files:

        RAW DATA a:lraw
   CHANNEL TABLE a:lcht
         ARCHIVE a:lraw.ach
             LOG b:lraw.log

Archive delimiter is " ".
Missing or bad data values are replaced by the value -99.000 .

Line errors: are identified by their line number in the raw data file.
Data errors: are identified by the channels name, line and position
             within the case: "name   "(line in case/position in line).
Line numbers in raw data file are shown as |number| or as *number*
             |numbers| indicates a line of data, *numbers* is a comment line.


First case on raw data: 86 009 11:20
-------------------------------------------------------------------------------
BeginDate: 86 009 12:20                            First output case: 86 009 12:40
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Data Error:
|17| 86009 1240    2.48    0.00    0.00    1497    0.00    0.00    0.00    0.00
 0.00    0.00    0.00    0.00
Value out of bounds: "T-hot   "(1/10); "T-ret   "(1/12);

Data Error:
|18| 86009 1300    2.48    0.00  248.00   11656    0.00    0.00    0.00    0.00
 0.00    0.00    0.00    0.00
Value out of bounds: "G-cook  "(1/5); "Elect   "(1/6); "T-hot   "(1/10); "T-ret
"(1/12);

Data Error:
|19| 86009 1320    3.02  158.00    1016   21807    0.00    0.00    0.00    0.00
 0.00    0.00    0.00    0.00
Value out of bounds: "G-main  "(1/4); "G-cook  "(1/5); "Elect   "(1/6); "T-hot
"(1/10); "T-ret   "(1/12);

Data Error:
|20| 86009 1340    3.57  383.00    1016   32029    0.00    0.00    0.00    0.00
 0.00    0.00    0.00    0.00
Value out of bounds: "G-main  "(1/4); "Elect   "(1/6); "T-hot   "(1/10); "T-ret
"(1/12);

Data Error:
|21| 86009 1400    3.57  686.00    1016   41963    0.00    0.00    0.00    0.00
 0.00    0.00    0.00    0.00
Value out of bounds: "G-main  "(1/4); "Elect   "(1/6); "T-hot   "(1/10); "T-ret
"(1/12);
```

```
Data Error:
|22| 86009 1440    0.00  686.00    1016   41963    0.00     0.00     0.00     0.00
 0.00    0.00    0.00    0.00
Value out of bounds: "T-hot   "(1/10); "T-ret   "(1/12);

Data Error:
|23| 86009 1500    4.65  737.00    2080   52086    0.00     0.00     0.00     0.00
 0.00    0.00    0.00    0.00
Value out of bounds: "G-cook  "(1/5); "Elect   "(1/6); "T-hot   "(1/10); "T-ret
"(1/12);

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
EndDate:   86 009 15:15                        Last output case: 86 009 15:00
--------------------------------------------------------------------------------

STATISTICS:

  24 lines read from beginning of raw data file.
   7 lines processed between Begin and End dates.
    (including 0 comments and 0 all-blank lines)
   0 line errors detected.


  25 data errors, and 4 missing data detected, itemized below:

  Arc  Channel   Error  Missing
  pos   name     count   count
   5   G-main      3       1
   6   G-cook      3       1
   7   Elect       5       1
   8   T-hot       7       0
   9   T-ret       7       0
  10   Q-cold      0       1
```

A.6 Practical example for raw data with multiple, numbered lines per case.

Raw data file:

```
S
Sending data for b:0525S;  Hit <Esc> to return to main menu.




 0 87145 0100 275
 1    TOA-S  TMER
 1   16.69   23.58  0
 3    N 08   N 10   N 12   N 14
 3    0.00      0.00   0.00    0.00  0
 4    N 03   N 09   N 11   N 13   N 15   N 17   N 18   N 20   N 24   N 25   N 32
 4  0.0000   0.00   0.00   0.00   0.00   0.00   0.00   0.77   0.56   4.48    6.28  0
 6    N 06   N 07   N 27   N 28   N 29   N 30   N 31   N 33
 6   99.96  12.10  26.01  27.23  26.85  26.24  25.89  16.08  0
 7    N 46   N 47   N 48
 7   22.70  27.19  10.55  0
 8    N 16
 8    0.00  0
 9    S 13   S 14   S 15   S 16   S 17   S 24   S 26   S 28   S 30   S 31   S 32   S 35   S 46
 9  0.0000 0.0000 0.0036 0.0000 0.0000 0.03   -0.08  -0.02  -0.00  -0.22   0.42  -0.01    1.58  0
10    S 19   MIX1   COIL1  MIX2   COIL2
10    0.02  17.69  19.91  17.28  18.40  0
11    S 02   S 03   S 04   S 05   S 07   S 08   S 09   S 10   S 11   S 12   S 18   S 20   S 21
11   22.40  22.52  24.22  24.53  23.20  23.19  24.37  24.50 999.00 999.00  23.62   22.94   22.29  0
12    S 22   TRET2  S 41   S 42   S 43   S 44   S 45   S 47   S 48   S 49   S 50
12   23.20  21.04  99.96  24.52  25.13  25.03  99.96  21.02 -100.00  22.64  24.42  0
13    S 23   S 25   S 27   S 29
13    0.00   0.00   0.00   0.00  0
14    S 01   S 06   S 33   S 40
14    2.02   2.02  16.42  69.12  0
15    N 01   N 02   N 50   N 53
15    1.73   0.58 171.36   6.91  0
16    N 54   N 55   N 56   N 57
16    8.21   1.15   3.02   4.61  0
17    N 19   N 58   N 59   N 60
17    2.74   7.20   1.73  16.13  0
 0 87145 0200 276
 1   16.04   23.50  0
 3    0.00      0.00   0.00    0.00  0
 4  0.0000   0.00   0.00   0.00   0.00   0.00   0.00   0.77   0.57   4.72    6.18  0
 6   99.96  12.41  25.93  27.15  26.91  26.18  25.88  15.15  0
 7   22.64  26.95  11.08  0
 8    0.00  0
 9  0.0000 0.0000 0.0036 0.0000 0.0000 0.03   -0.08  -0.02  -0.00  -0.29   0.33  -0.01    1.58  0
10    0.02  17.19  19.86  16.67  18.11  0
11   22.46  22.56  24.22  24.52  23.23  23.19  24.37  24.50 999.00 999.00  23.63   22.95   22.33  0
12   23.24  20.75  24.53  24.34  24.99  24.85  23.80  20.27 -100.00  22.45  24.34  0
13    0.00   0.00   0.00   0.00  0
14    2.02   2.16  16.56  69.12  0
15    1.30   1.01 1777.1   7.06  0
16    8.06   1.01   3.02   4.61  0
17    2.88   8.06   1.58  15.55  0
 0 87145 0300 248
 1   15.57   23.12  0
 3    0.00      0.00   0.00    0.00  0
 4  0.0000   0.00   0.00   0.00   0.00   0.00   0.00   0.78   0.54   4.73    6.06  0
 6   99.86  12.70  25.84  27.05  26.56  26.13  25.85  14.38  0
 7   22.32  26.80  11.60  0
 8    0.00  0
 9  0.1774 0.6613 0.6169 0.0000 0.0000 17.32  17.74  10.33  14.48  19.51  17.26    1.76    1.58  0
10    4.47  20.49  18.39  14.06  13.67  0
11   16.49  19.03  23.93  27.63  23.09  23.20  24.35  24.48   4.71   0.19  15.01   14.97   18.77  0
12   19.25  22.22  23.22  23.83  24.19  24.38  22.52  20.00 -100.00  23.66  20.50  0
13   28.66  38.45   9.22  10.08  0
14   18.58   2.02  15.98 211.68  0
15    1.73   0.86 1777.1   7.06  0
16    7.78   1.44   3.02   3.74  0
17    3.17   7.63   1.73  15.55  0
```

```
 0 87145 0400 242
 1   14.72   22.99  0
 3   ^@.44      1.58    0.43    0.58  0
 4  0.0539    0.21    0.26    0.21    0.27    2.04    0.09   -2.91   -0.50    4.59    6.05  0
 6   99.96   13.11   25.74   26.93   26.34   26.06   25.77   13.61  0
 7   22.09   26.61   12.18  0
 8    0.14  0
 9  0.4215  1.0000  0.6405  0.0000  0.0000 16.90   18.07   11.01   14.20   20.86   17.48    1.90    1.58  0
10    4.73   20.66   17.66   13.11   13.09  0
11    9.47   13.82   23.74   29.08   20.19   23.21   24.35   24.27    1.49    3.02   13.08   13.61   14.28  0
12   13.73   21.77   21.48   23.44   24.10   23.88   22.23   19.85 -100.00   23.25   19.62  0
13   29.81   37.87    9.65   10.37  0
14   12.24    2.16   16.13  211.68  0
15    2.30    1.44  187.20    7.06  0
16    8.78    1.73    4.61    4.03  0
17    3.31    8.21    4.18   16.70  0
.
.
.
.
.   (part of raw data file removed)
.
.
.
.
.
 0 87145 2300 284
 1   14.85   23.02  0
 3    0.00      0.00    0.00    0.00  0
 4  0.0000    0.00    0.00    0.00    0.00    0.00    0.00    0.59    0.49    3.20    5.12  0
 6   99.96    8.58   23.39   25.07   23.77   24.24   23.28   14.46  0
 7   18.31   24.66    8.08  0
 8    0.00  0
 9  0.0035  0.0000  0.0000  0.0000  0.0000  0.03   -0.02   -0.00   -0.00   -0.34    0.35   -0.01    1.58  0
10    0.02   16.47   18.53   14.94   16.07  0
11   10.17   15.26   27.15   27.03   18.04   22.91   23.78   23.66    5.03  999.00   17.04   15.64   14.99  0
12   15.76   19.45   21.78   23.40   24.11   23.13   22.22   20.18 -100.00   22.48   23.95  0
13    0.00    0.00    0.00    0.00  0
14    2.02    2.02   73.44  132.48  0
15    1.58    0.72  227.52   46.66  0
16    8.93    2.02    3.02    4.18  0
17  [illegible]  9.63    1.73   17.86  0
 0 87145 2400 283
 1   14.29   22.95  0
 3    0.00      0.00    0.00    0.00  0
 4  0.0000    0.00    0.00    0.00    0.00    0.00    0.00    0.56    0.49    2.12    5.07  0
 6   99.96    8.81   23.39   25.06   23.96   24.27   23.37   13.28  0
 7   18.42   24.51    8.58  0
 8    0.00  0
 9  0.0000  0.0000  0.0000  0.0000  0.0000  0.03   -0.02   -0.00   -0.00   -0.34    0.21   -0.01    1.58  0
10    0.02   16.08   18.61   14.56   15.96  0
11   10.72   15.63   26.70   26.70   18.17   22.87   23.77   23.72  999.00  999.00   17.53   16.02   15.26  0
12   16.18   19.58   21.74   23.24   24.66   22.99   22.09   19.32 -100.00   22.37   23.86  0
13    0.00    0.00    0.00    0.00  0
14    2.02    2.16   41.76  102.24  0
15    2.02    1.30  204.48   23.76  0
16    9.07    2.16    3.02    4.61  0
17    5.04    9.94    1.73   17.86  0

Hit <↵> to return to main menu...
```

Channel table:

Channel Table: ENERPLEX.CHT (summer season).

Date mm/dd/yy	Time hh:mm	Lin	Raw pos	ACH pos	Name of channel	Archive Units	ACH Format	Convers. and calib. Code	Constants:	Error checking Code	Constants:	Channel Description
#25(9999)6(,)												
04/25/87	13:00	0	0	0	Begin	ENERPLEX						
04/25/87	13:00	0	1	1	Date	yyddd	I6	20		0		Date
04/25/87	13:00	0	2	2	Time	hhmm	I5	15		0		Time
04/25/87	13:00	0	1	3	Reftime	ddddd.hhhh	F11.4	28		0		Reference time from Jan 1, 80
04/25/87	13:00	0	3	4	Polled	#/period	I4	1		0		Number of times polled
04/25/87	13:00	1	1	55	TOA-S	C	F6.2	1		0		AC1S Outdoor air intake temp
04/25/87	13:00	1	2	56	TMER	C	F6.2	1		0		Mechanical equip room temp (South)
04/25/87	13:00	1	3	0	S&C_F	hex	I1	1		0		Sampling/Cumulative flag (0-F)
04/25/87	13:00	3	1	63	N_08	kWh	F6.2	1		0		Supply fan 1 electricity
04/25/87	13:00	3	2	65	N_10	kWh	F6.2	1		0		Supply fan 2 electricity
04/25/87	13:00	3	3	67	N_12	kWh	F6.2	1		0		Return fan 1 electricity
04/25/87	13:00	3	4	69	N_14	kWh	F6.2	1		0		Return fan 2 electricity
04/25/87	13:00	3	5	0	S&C_F	hex	I1	1		0		Sampling/Cumulative flag (0-F)
04/25/87	13:00	4	1	59	N_03	hrs	F5.2	1		0		Chilled water pump 1 runtime
04/25/87	13:00	4	2	64	N_09	m3/s	F6.2	1		0		Supply fan 1 air flow rate
04/25/87	13:00	4	3	66	N_11	m3/s	F6.2	1		0		Supply fan 2 air flow rate
04/25/87	13:00	4	4	68	N_13	m3/s	F6.2	1		0		Return fan 1 air flow rate
04/25/87	13:00	4	5	70	N_15	m3/s	F6.2	1		0		Return fan 2 air flow rate
04/25/87	13:00	4	6	72	N_17	m3/s	F6.2	1		0		Plenum 1 recirculated air flow
04/25/87	13:00	4	7	73	N_18	m3/s	F6.2	1		0		Plenum 2 recirculated air flow
04/25/87	13:00	4	8	75	N_20	W/m2	F8.2	1		0		Solar radiation roof
04/25/87	13:00	4	9	76	N_24	W/m2	F7.2	1		0		Diffuse solar radiation (shadow band)
04/25/87	13:00	4	10	77	N_25	m/s	F6.2	1		0		Wind speed roof
04/25/87	13:00	4	11	83	N_32	%	F7.2	1		0		Interior relative humidity
04/25/87	13:00	4	12	0	S&C_F	hex	I1	1		0		Sampling/Cumulative flag (0-F)
04/25/87	13:00	6	9	60	N_05	gpm	F7.2	0		0		Chilled water flow rate
04/25/87	13:00	6	1	61	N_06	C	F6.2	1		0		HX building side outlet temp
04/25/87	13:00	6	2	62	N_07	C	F6.2	1		0		HX building side inlet temp
04/25/87	13:00	6	3	78	N_27	C	F6.2	1		1	5 50	East zone temperature
04/25/87	13:00	6	4	79	N_28	C	F6.2	1		1	5 50	West zone temperature
04/25/87	13:00	6	5	80	N_29	C	F6.2	1		1	5 50	North zone temperature
04/25/87	13:00	6	6	81	N_30	C	F6.2	1		1	5 50	South zone temperature
04/25/87	13:00	6	7	82	N_31	C	F6.2	1		1	5 50	Interior zone temperature
04/25/87	13:00	6	8	84	N_33	C	F7.2	1		0		Ambient dry buld temperature
04/25/87	13:00	6	9	0	S&C_F	hex	I1	1		0		Sampling/Cumulative flag (0-F)
04/25/87	13:00	7	1	85	N_46	C	F6.2	1		0		Supply air temperature
04/25/87	13:00	7	2	86	N_47	C	F6.2	1		0		Return air temperature
04/25/87	13:00	7	3	87	N_48	C	F6.2	1		0		Chillers out temperature
04/25/87	13:00	7	4	0	S&C_F	hex	I1	1		0		Sampling/Cumulative flag (0-F)
04/25/87	13:00	8	1	71	N_16	kWh	F6.2	1		0		Double wall fan electricity
04/25/87	13:00	8	2	0	S&C_F	hex	I1	1		0		Sampling/Cumulative flag (0-F)
04/25/87	13:00	9	1	17	S_13	hrs	F5.2	1		0		Condenser water pump 1 runtime
04/25/87	13:00	9	2	18	S_14	hrs	F5.2	1		0		Condenser water pump 2 runtime
04/25/87	13:00	9	3	19	S_15	hrs	F5.2	1		0		Ground water pump 1 runtime
04/25/87	13:00	9	4	20	S_16	hrs	F5.2	1		0		Ground water pump 2 runtime
04/25/87	13:00	9	5	21	S_17	hrs	F5.2	1		0		Ground water pump 3 runtime
04/25/87	13:00	9	6	28	S_24	cfm	F6.2	2	2119 0	1	0 60000	Supply fan 1 air flow rate
04/25/87	13:00	9	7	30	S_26	cfm	F6.2	2	2119 0	1	0 60000	Supply fan 2 air flow rate
04/25/87	13:00	9	8	32	S_28	cfm	F6.2	2	2119 0	1	0 60000	Return fan 1 air flow rate
04/25/87	13:00	9	9	34	S_30	cfm	F6.2	2	2119 0	1	0 60000	Return fan 2 air flow rate
04/25/87	13:00	9	10	35	S_31	cfm	F6.2	2	2119 0	0		Spill air 1 air flow rate
04/25/87	13:00	9	11	36	S_32	cfm	F6.2	2	2119 0	0		Spill air 2 air flow rate
04/25/87	13:00	9	12	38	S_35	cfm	F6.2	2	2119 0	0		Return fan 2 relief air flow
04/25/87	13:00	9	13	45	S_46	cfm	F6.2	2	2119 0	0		Return fan 1 relief air flow
04/25/87	13:00	9	14	0	S&C_F	hex	I1	1		0		Sampling/Cumulative flag (0-F)
04/25/87	13:00	10	1	23	S_19	kg/s	F7.2	1		0		Well water flow rate
04/25/87	13:00	10	2	51	MIX1	C	F6.2	1		0		West return/outdoor mix air temp (South)
04/25/87	13:00	10	3	50	COIL1	C	F6.2	1		0		AC1S supply air temperature (upstream)
04/25/87	13:00	10	4	54	MIX2	C	F6.2	1		0		East return/outdoor mix air temp (South)
04/25/87	13:00	10	5	52	COIL2	C	F6.2	1		0		AC2S supply air temperature (upstream)
04/25/87	13:00	10	6	0	S&C_F	hex	I1	1		0		Sampling/Cumulative flag (0-F)
04/25/87	13:00	11	1	6	S_02	C	F6.2	1		0		HP1 evaporator outlet temperature
04/25/87	13:00	11	2	7	S_03	C	F6.2	1		0		HP1 evaporator inlet temperature
04/25/87	13:00	11	3	8	S_04	C	F6.2	1		0		HP1 condenser inlet temperature
04/25/87	13:00	11	4	9	S_05	C	F6.2	1		0		HP1 condenser outlet temperature
04/25/87	13:00	11	5	11	S_07	C	F6.2	1		0		HP2 evaporator outlet temperature
04/25/87	13:00	11	6	12	S_08	C	F6.2	1		0		HP2 evaporator inlet temperature

04/25/87 13:00	11	7	13	S_09	C	F6.2	1	0		HP2 condenser inlet temperature
04/25/87 13:00	11	8	14	S_10	C	F6.2	1	0		HP2 condenser outlet temperature
04/25/87 13:00	11	9	15	S_11	C	F6.2	1	0		HP1 evaporator delta
04/25/87 13:00	11	10	16	S_12	C	F6.2	1	0		HP2 evaporator delta
04/25/87 13:00	11	11	22	S_18	C	F6.2	1	0		HX well water inlet temperature
04/25/87 13:00	11	12	24	S_20	C	F6.2	1	0		HX well water outlet temperature
04/25/87 13:00	11	13	25	S_21	C	F6.2	1	0		HX building side inlet temperature
04/25/87 13:00	11	14	0	S&C_F	hex	I1	1	0		Sampling/Cumulative flag (0-F)
04/25/87 13:00	12	1	26	S_22	C	F6.2	1	0		HX building side outlet temperature
04/25/87 13:00	12	2	53	TRET2	C	F6.2	1	0		RF2S return air temperature (downstream)
04/25/87 13:00	12	3	40	S_41	C	F6.2	1	1	5 50	East zone temperature
04/25/87 13:00	12	4	41	S_42	C	F6.2	1	1	5 50	West zone temperature
04/25/87 13:00	12	5	42	S_43	C	F6.2	1	1	5 50	North zone temperature
04/25/87 13:00	12	6	43	S_44	C	F6.2	1	1	5 50	South zone temperature
04/25/87 13:00	12	7	44	S_45	C	F6.2	1	1	5 50	Interior zone temperature
04/25/87 13:00	12	8	46	S_47	C	F6.2	1	0		Atrium top temperature
04/25/87 13:00	12	9	47	S_48	C	F6.2	1	0		Atrium bottom temperature
04/25/87 13:00	12	10	48	S_49	C	F6.2	1	0		RF1S return air temperature (upstream)
04/25/87 13:00	12	11	49	S_50	C	F6.2	1	0		AC1S supply air temperature (downstream)
04/25/87 13:00	12	12	0	S&C_F	hex	I1	1	0		Sampling/Cumulative flag (0-F)
04/25/87 13:00	13	1	27	S_23	kWh	F6.2	1	0		Supply fan 1 electricity
04/25/87 13:00	13	2	29	S_25	kWh	F6.2	1	0		Supply fan 2 electricity
04/25/87 13:00	13	3	31	S_27	kWh	F6.2	1	0		Return fan 1 electricity
04/25/87 13:00	13	4	33	S_29	kWh	F6.2	1	0		Return fan 2 electricity
04/25/87 13:00	13	5	0	S&C_F	hex	I1	1	0		Sampling/Cumulative flag (0-F)
04/25/87 13:00	14	1	5	S_01	kWh	F7.2	1	0		Heat pump 1 electricity
04/25/87 13:00	14	2	10	S_06	kWh	F7.2	1	0		Heat pump 2 electricity
04/25/87 13:00	14	3	37	S_33	kWh	F7.2	1	0		House panel electricity
04/25/87 13:00	14	4	39	S_40	kWh	F8.2	1	1	0 1200	Total building electricity
04/25/87 13:00	14	5	0	S&C_F	hex	I1	1	0		Sampling/Cumulative flag (0-F)
04/25/87 13:00	15	1	57	N_01	kWh	F7.2	1	0		Chiller 1 electricity
04/25/87 13:00	15	2	58	N_02	kWh	F7.2	1	0		Chiller 2 electricity
04/25/87 13:00	15	3	88	N_50	kWh	F8.2	1	1	0 1200	Total building electricity
04/25/87 13:00	15	4	89	N_53	kWh	F7.2	1	0		House panel electricity
04/25/87 13:00	15	5	0	S&C_F	hex	I1	1	0		Sampling/Cumulative flag (0-F)
04/25/87 13:00	16	1	90	N_54	kWh	F7.2	1	0		Riser 2-3N & 3-3N electricity
04/25/87 13:00	16	2	91	N_55	kWh	F7.2	1	0		Riser 3-4N electricity
04/25/87 13:00	16	3	92	N_56	kWh	F7.2	1	0		Riser G-2N electricity
04/25/87 13:00	16	4	93	N_57	kWh	F7.2	1	0		Riser 2-2N electricity
04/25/87 13:00	16	5	0	S&C_F	hex	I1	1	0		Sampling/Cumulative flag (0-F)
04/25/87 13:00	17	1	74	N_19	kWh	F7.2	1	0		Riser 3-2N electricity
04/25/87 13:00	17	2	94	N_58	kWh	F7.2	1	0		Riser 2-4N electricity
04/25/87 13:00	17	3	95	N_59	kWh	F7.2	1	0		Riser G-4N electricity
04/25/87 13:00	17	4	96	N_60	kWh	F7.2	1	0		Riser G-1N, 2-1N, & 3-1N electricity
04/25/87 13:00	17	5	0	S&C_F	hex	I1	1	0		Sampling/Cumulative flag (0-F)
12/31/99 00:00	0	0	0	End	gvs					

Archive file:

To fit on this page, each line is split into several lines below.

```
 87145,  100,  2701.0417, 275,   2.02, 22.40, 22.52, 24.22, 24.53,   2.02, 23.20, 23.19, 24.37, 24.50,999.00,999.00, 0.00, 0.00, 0.00,
0.00, 0.00, 23.62,   0.02, 22.94, 22.29, 23.20,  0.00, 63.57,  0.00,  9999,  0.00,  9999,  0.00,  0.00,-466.18,889.98,  16.42,-21.19,
69.12,  9999, 24.52, 25.13, 25.03,  9999,3348.02, 21.02,-100.00, 22.64, 24.42, 19.91, 17.69, 18.40, 21.04, 17.28, 16.69, 23.58,   1.73,
 0.58, 0.00,   9999, 99.96, 12.10,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,   2.74,    0.77,   0.56,
4.48, 26.01, 27.23, 26.85, 26.24, 25.89,   6.28,  16.08, 22.70, 27.19, 10.55,  171.36,   6.91,   8.21,   1.15,   3.02,   4.61,   7.20,
1.73,  16.13
 87145,  200,  2701.0833, 276,   2.02, 22.46, 22.56, 24.22, 24.52,   2.16, 23.23, 23.19, 24.37, 24.50,999.00,999.00, 0.00, 0.00, 0.00,
0.00, 0.00, 23.63,   0.02, 22.95, 22.33, 23.24,  0.00, 63.57,  0.00,  9999,  0.00,  9999,  0.00,  0.00,-614.51,699.27,  16.56,-21.19,
69.12, 24.53, 24.34, 24.99, 24.85, 23.80,3348.02, 20.27,-100.00, 22.45, 24.34, 19.86, 17.19, 18.11, 20.75, 16.67, 16.04, 23.50,   1.30,
 1.01, 0.00,   9999, 99.96, 12.41,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,   2.88,    0.77,   0.57,
4.72, 25.93, 27.15, 26.91, 26.18, 25.88,   6.18,  15.15, 22.64, 26.95, 11.08,    9999,   7.06,   8.06,   1.01,   3.02,   4.61,   8.06,
1.58,  15.55
 87145,  300,  2701.1250, 248,  18.58, 16.49, 19.03, 23.93, 27.63,   2.02, 23.09, 23.20, 24.35, 24.48,  4.71,  0.19, 0.18, 0.66, 0.62,
0.00, 0.00, 15.01,   4.47, 14.97, 18.77, 19.25, 28.66,36701.08, 38.45,37591.06,  9.22,21889.27, 10.08,30683.12,41341.69,36573.94,
15.98,3729.44,  211.68, 23.22, 23.83, 24.19, 24.38, 22.52,3348.02, 20.00,-100.00, 23.66, 20.50, 18.39, 20.49, 13.67, 22.22, 14.06,
15.57, 23.12,   1.73,   0.86, 0.00,   9999, 99.96, 12.70,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,  0.00,
3.17,    0.78,   0.54,  4.73, 25.84, 27.05, 26.56, 26.13, 25.85,   6.06,  14.38, 22.32, 26.80, 11.60,    9999,   7.06,   7.78,   1.44,
3.02,   3.74,   7.63,   1.73,  15.55
```

```
87145, 400, 2701.1667, 242, 12.24, 9.47, 13.82, 23.74, 29.08, 2.16, 20.19, 23.21, 24.35, 24.27, 1.49, 3.02, 0.42, 1.00, 0.64,
0.00, 0.00, 13.08, 4.73, 13.61, 14.28, 13.73, 29.81,35811.10, 37.87,38290.33, 9.65,23330.19, 10.37,30089.80,44202.34,37040.12,
16.13,4026.10, 211.68, 21.48, 23.44, 24.10, 23.88, 22.23,3348.02, 19.85,-100.00, 23.25, 19.62, 17.66, 20.66, 13.09, 21.77, 13.11,
14.72, 22.99, 2.30, 1.44, 0.05, 9999, 99.96, 13.11, 9999, 0.21, 1.58, 0.26, 0.43, 0.21, 0.58, 0.27, 0.14, 2.04, 0.09,
3.31, -2.91, -0.50, 4.59, 25.74, 26.93, 26.34, 26.06, 25.77, 6.05, 13.61, 22.09, 26.61, 12.18, 187.20, 7.06, 8.78, 1.73,
4.61, 4.03, 8.21, 4.18, 16.70
.
.
.
.
.    (part of archive file removed)
.
.
.
.
87145, 2300, 2701.9583, 284, 2.02, 10.17, 15.26, 27.15, 27.03, 2.02, 18.04, 22.91, 23.78, 23.66, 5.03,999.00, 0.00, 0.00, 0.00,
0.00, 0.00, 17.04, 0.02, 15.64, 14.99, 15.76, 0.00, 63.57, 0.00, 9999, 0.00, 0.00, 0.00, 0.00,-720.46,741.65, 73.44,-21.19,
132.48, 21.78, 23.40, 24.11, 23.13, 22.22,3348.02, 20.18,-100.00, 22.48, 23.95, 18.53, 16.47, 16.07, 19.45, 14.94, 14.85, 23.02, 1.58,
0.72, 0.00, 9999, 99.96, 8.58, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 6.05, 0.59, 0.49,
3.20, 23.39, 25.07, 23.77, 24.24, 23.28, 5.12, 14.46, 18.31, 24.66, 8.08, 227.52, 46.66, 8.93, 2.02, 3.02, 4.18, 9.65,
1.73, 17.86
87145, 2400, 2702.0000, 283, 2.02, 10.72, 15.63, 26.70, 26.70, 2.16, 18.17, 22.87, 23.77, 23.72,999.00,999.00, 0.00, 0.00, 0.00,
0.00, 0.00, 17.53, 0.02, 16.02, 15.26, 16.18, 0.00, 63.57, 0.00, 9999, 0.00, 0.00, 0.00, 0.00,-720.46,444.99, 41.76,-21.19,
102.24, 21.74, 23.24, 24.66, 22.99, 22.09,3348.02, 19.32,-100.00, 22.37, 23.86, 18.61, 16.08, 15.96, 19.58, 14.56, 14.29, 22.95, 2.02,
1.30, 0.00, 9999, 99.96, 8.81, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 5.04, 0.56, 0.49,
2.12, 23.39, 25.06, 23.96, 24.27, 23.37, 5.07, 13.28, 18.42, 24.51, 8.58, 204.48, 23.76, 9.07, 2.16, 3.02, 4.61, 9.94,
1.73, 17.86
```

Log file:

```
Log of Archive, version: 1.4 of 1 June 1987,     processed on  9 Jun 1987

Files:

        RAW DATA a:e870525s.raw
   CHANNEL TABLE a:enerplex.cht
         ARCHIVE a:e870525s.ach
             LOG a:e870525s.log


Archive delimiter is ",".
Missing or bad data values are replaced by the string "9999".

Line errors: are identified by their line number in the raw data file.
Data errors: are identified by the channels name, line and position
             within the case: "name  "(line in case/position in line).
Line numbers in raw data file are shown as |number| or as *number*
             |numbers| indicates a line of data, *numbers* is a comment line.


*1* S

*2* Sending data for b:0525S;  Hit <Esc> to return to main menu.

First case on raw data: 87 145 01:00
```

```
--------------------------------------------------------------------------------
BeginDate: 87 115 13:00                          First output case: 87 145 01:00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

*15*  1    TOA-S  TMER

*17*  3    N 08   N 10   N 12   N 14

*19*  4    N 03   N 09   N 11   N 13   N 15   N 17   N 18   N 20   N 24   N 25
N 32

*21*  6    N 06   N 07   N 27   N 28   N 29   N 30   N 31   N 33

*23*  7    N 46   N 47   N 48

*25*  8    N 16

*27*  9    S 13   S 14   S 15   S 16   S 17   S 24   S 26   S 28   S 30   S 31
S 32   S 35   S 46

*29*  10   S 19   MIX1   COIL1  MIX2   COIL2

*31*  11   S 02   S 03   S 04   S 05   S 07   S 08   S 09   S 10   S 11   S 12
S 18   S 20   S 21

*33*  12   S 22   TRET2  S 41   S 42   S 43   S 44   S 45   S 47   S 48   S 49
S 50

*35*  13   S 23   S 25   S 27   S 29

*37*  14   S 01   S 06   S 33   S 40

*39*  15   N 01   N 02   N 50   N 53

*41*  16   N 54   N 55   N 56   N 57

*43*  17   N 19   N 58   N 59   N 60

Data Error:
|14|   0 87145 0100 275
|16|   1   16.69    23.58  0
|18|   3    0.00       0.00     0.00     0.00  0
|20|   4  0.0000    0.00     0.00     0.00     0.00     0.00     0.00     0.77     0.56
  4.48     6.28  0
|22|   6   99.96    12.10    26.01    27.23    26.85    26.24    25.89    16.08  0
|24|   7   22.70    27.19    10.55  0
|26|   8    0.00  0
|28|   9  0.0000  0.0000  0.0036  0.0000  0.0000  0.03    -0.08    -0.02    -0.00
-0.22     0.42   -0.01     1.58  0
|30|  10    0.02    17.69    19.91    17.28    18.40  0
|32|  11   22.40    22.52    24.22    24.53    23.20    23.19    24.37    24.50  999.00
999.00   23.62    22.94    22.29  0
|34|  12   23.20    21.04    99.96    24.52    25.13    25.03    99.96    21.02 -100.00
 22.64    24.42  0
```

```
|36|  13     0.00     0.00     0.00     0.00  0
|38|  14     2.02     2.02    16.42    69.12  0
|40|  15     1.73     0.58   171.36     6.91  0
|42|  16     8.21     1.15     3.02     4.61  0
|44|  17     2.74     7.20     1.73    16.13  0
Value out of bounds: "S_26    "(9/7); "S_28    "(9/8); "S_41    "(12/3); "S_45
"(12/7);

Data Error:
|45|    0 87145 0200 276
|46|    1   16.04    23.50  0
|47|    3    0.00      0.00     0.00     0.00  0
|48|    4  0.0000    0.00     0.00     0.00     0.00     0.00     0.00     0.77     0.57
  4.72     6.18  0
|49|    6   99.96    12.41    25.93    27.15    26.91    26.18    25.88    15.15  0
|50|    7   22.64    26.95    11.08  0
|51|    8    0.00  0
|52|    9  0.0000  0.0000  0.0036  0.0000  0.0000  0.03    -0.08    -0.02    -0.00
-0.29     0.33    -0.01     1.58  0
|53|  10     0.02    17.19    19.86    16.67    18.11  0
|54|  11    22.46    22.56    24.22    24.52    23.23    23.19    24.37    24.50  999.00
999.00    23.63    22.95    22.33  0
|55|  12    23.24    20.75    24.53    24.34    24.99    24.85    23.80    20.27 -100.00
 22.45    24.34  0
|56|  13     0.00     0.00     0.00     0.00  0
|57|  14     2.02     2.16    16.56    69.12  0
|58|  15     1.30     1.01   1777.1     7.06  0
|59|  16     8.06     1.01     3.02     4.61  0
|60|  17     2.88     8.06     1.58    15.55  0
Value out of bounds: "S_26    "(9/7); "S_28    "(9/8); "N_50    "(15/3);

Data Error:
|61|    0 87145 0300 248
|62|    1   15.57    23.12  0
|63|    3    0.00      0.00     0.00     0.00  0
|64|    4  0.0000    0.00     0.00     0.00     0.00     0.00     0.00     0.78     0.54
  4.73     6.06  0
|65|    6   99.96    12.70    25.84    27.05    26.56    26.13    25.85    14.38  0
|66|    7   22.32    26.80    11.60  0
|67|    8    0.00  0
|68|    9  0.1774  0.6613  0.6169  0.0000  0.0000 17.32    17.74    10.33    14.48
19.51    17.26     1.76     1.58  0
|69|  10     4.47    20.49    18.39    14.06    13.67  0
|70|  11    16.49    19.03    23.93    27.63    23.09    23.20    24.35    24.48     4.71
  0.19    15.01    14.97    18.77  0
|71|  12    19.25    22.22    23.22    23.83    24.19    24.38    22.52    20.00 -100.00
 23.66    20.50  0
|72|  13    28.66    38.45     9.22    10.08  0
|73|  14    18.58     2.02    15.98   211.68  0
|74|  15     1.73     0.86   1777.1     7.06  0
|75|  16     7.78     1.44     3.02     3.74  0
|76|  17     3.17     7.63     1.73    15.55  0
Value out of bounds: "N_50    "(15/3);
```

```
Data Error:
|77|   0 87145 0400 242
|78|   1   14.72    22.99  0
|79|   3   ^@.44       1.58     0.43     0.58  0
|80|   4  0.0539    0.21    0.26    0.21    0.27    2.04    0.09   -2.91   -0.50
  4.59    6.05  0
|81|   6   99.96   13.11   25.74   26.93   26.34   26.06   25.77   13.61  0
|82|   7   22.09   26.61   12.18  0
|83|   8    0.14  0
|84|   9  0.4215  1.0000  0.6405  0.0000  0.0000 16.90   18.07   11.01   14.20
20.86   17.48    1.90    1.58  0
|85|  10    4.73   20.66   17.66   13.11   13.09  0
|86|  11    9.47   13.82   23.74   29.08   20.19   23.21   24.35   24.27    1.49
  3.02   13.08   13.61   14.28  0
|87|  12   13.73   21.77   21.48   23.44   24.10   23.88   22.23   19.85 -100.00
 23.25   19.62  0
|88|  13   29.81   37.87    9.65   10.37  0
|89|  14   12.24    2.16   16.13  211.68  0
|90|  15    2.30    1.44  187.20    7.06  0
|91|  16    8.78    1.73    4.61    4.03  0
|92|  17    3.31    8.21    4.18   16.70  0
bad number: "N_08    "(3/1);
.
.
.                     (part of the log file removed)
.
.
Data Error:
|382|   0 87145 2300 284
|383|   1   14.85   23.02  0
|384|   3    0.00      0.00    0.00    0.00  0
|385|   4  0.0000   0.00    0.00    0.00    0.00    0.00    0.00    0.59    0.49
  3.20    5.12  0
|386|   6   99.96    8.58   23.39   25.07   23.77   24.24   23.28   14.46  0
|387|   7   18.31   24.66    8.08  0
|388|   8    0.00  0
|389|   9  0.0035  0.0000  0.0000  0.0000  0.0000  0.03   -0.02   -0.00   -0.00
-0.34    0.35   -0.01    1.58  0
|390|  10    0.02   16.47   18.53   14.94   16.07  0
|391|  11   10.17   15.26   27.15   27.03   18.04   22.91   23.78   23.66    5.03
999.00   17.04   15.64   14.99  0
|392|  12   15.76   19.45   21.78   23.40   24.11   23.13   22.22   20.18 -100.00
 22.48   23.95  0
|393|  13    0.00    0.00    0.00    0.00  0
|394|  14    2.02    2.02   73.44  132.48  0
|395|  15    1.58    0.72  227.52   46.66  0
|396|  16    8.93    2.02    3.02    4.18  0
|397|  17    6.05    9.65    1.73   17.86  0
Value out of bounds: "S_26    "(9/7);

*415*  Hit <M> to return to main menu...

Data Error:
|398|   0 87145 2400 283
```

```
|399|  1   14.29    22.95  0
|400|  3    0.00       0.00    0.00    0.00  0
|401|  4  0.0000    0.00    0.00    0.00    0.00    0.00    0.00    0.56    0.49
  2.12    5.07  0
|402|  6   99.96    8.81   23.39   25.06   23.96   24.27   23.37   13.28  0
|403|  7   18.42   24.51    8.58  0
|404|  8    0.00  0
|405|  9  0.0000  0.0000  0.0000  0.0000  0.0000  0.03   -0.02   -0.00   -0.00
-0.34    0.21   -0.01    1.58  0
|406| 10    0.02   16.08   18.61   14.56   15.96  0
|407| 11   10.72   15.63   26.70   26.70   18.17   22.87   23.77   23.72  999.00
999.00   17.53   16.02   15.26  0
|408| 12   16.18   19.58   21.74   23.24   24.66   22.99   22.09   19.32 -100.00
 22.37   23.86  0
|409| 13    0.00    0.00    0.00    0.00  0
|410| 14    2.02    2.16   41.76  102.24  0
|411| 15    2.02    1.30  204.48   23.76  0
|412| 16    9.07    2.16    3.02    4.61  0
|413| 17    5.04    9.94    1.73   17.86  0
Value out of bounds: "S_26    "(9/7);

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
EndDate:   99 365 00:00                          Last output case: 87 145 24:00
--------------------------------------------------------------------------------
```

STATISTICS:

415 lines read from beginning of raw data file.
402 lines processed between Begin and End dates.
(including 16 comments and 1 all-blank lines)
1 line errors detected.

24 cases read; 24 cases archived.

31 data errors, and 24 missing data detected, itemized below:

Arc pos	Channel name	Error count	Missing count
17	S_13	1	0
18	S_14	1	0
28	S_24	1	0
30	S_26	10	0
32	S_28	7	0
34	S_30	1	0
39	S_40	1	0
40	S_41	1	0
44	S_45	1	0
60	N_05	0	24
62	N_07	1	0
63	N_08	3	0
78	N_27	1	0
88	N_50	2	0